Równoległe transformacje grafowe – podstawy formalne i zastosowania

Leszek Kotulski, Artur Basiura, Sebastian Ernst, Adam Sędziwy, Igor Wojnicki

AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, Kraków

Streszczenie: W artykule przedstawiono historię rozwoju i wykorzystania mechanizmu transformacji grafowych w ramach prac grupy badawczej GRADIS w latach 2006–2022 na Wydziale Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej. W ich efekcie powstał szereg międzynarodowych publikacji, nastąpił rozwój naukowy (stopnie profesorskie, habilitacje i doktoraty) oraz wdrożono wiele praktycznych rozwiązań w ramach komercjalizacji wyników naukowych.

Słowa kluczowe: gramatyki grafowe, równoległe transformacje grafowe, wspomaganie projektowania złożonych systemów

PARALLEL GRAPH TRANSFORMATION – FORMAL BACKGROUND AND APPLICATIONS

Abstract: This paper presents the history of development and application of the graph transformations mechanism, implemented by the GRADIS research group in the period of 2006–2022 at the Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering. These works resulted in numerous internationally-recognized publications, scientific development (professor, D.Sc. and Ph.D. degrees) and many practical and commercial applications of the results.

Keywords: graph grammars, parallel graph transformations, design support of complex systems

https://doi.org/10.7494/978-83-67427-00-5_5

1. Wstęp

Transformacje grafowe są mechanizmem formalnym wykorzystywanym w wielu zastosowaniach. Doskonałym podsumowaniem kilkudziesięciu lat badań w tej dziedzinie jest trzytomowa monografia pod redakcją Grzegorza Rozenberga ze współedytorami (Rozenberg 1997, Ehrig i in. 1999a, 1999b) oraz nieco późniejsza monografia Hartmuta Ehriga (Ehrig i in. 2015). W kontekście prac wykonywanych w Akademii Górniczo-Hutniczej (AGH) można wyróżnić dwa przenikające się kierunki działań: jednym z nich jest rozwój mechanizmów (zarówno pod względem ich efektywności, jak i siły ekspresji), drugim jest wykazanie użyteczności wypracowanych rozwiązań w praktyce.

Chronologicznie można tutaj wyróżnić następujące etapy:

1. Wprowadzenie koncepcji równoległych transformacji grafowych w celu podniesienia efektywności rozwiązań opartych na transformacjach (patrz rozdz. 3).
2. Badanie siły ekspresji różnych typów transformacji grafowych w aspekcie wspomagania projektowania architektonicznego (patrz rozdz. 3).
3. Wprowadzenie koncepcji gramatyk dualnych w celu podniesienia siły ekspresji w specyfikacji problemu (patrz rozdz. 4).

Kolejnym krokiem podnoszącym efektywność transformacji grafowych było zdefiniowanie pojęcia kontekstu przetwarzania i mechanizmu opóźnionego wykonywania produkcji (patrz rozdz. 5).

Aparat formalny transformacji grafowych precyzyjnie specyfikując problem, umożliwia efektywne poszukiwanie rozwiązań (patrz rozdz. 7) po uprzednim rozstrzygnięciu problemu generacji grafu na podstawie istniejących danych zewnętrznych. Koncepcja gramatyki grafowej, której produkcje wyzwalane są w wyniku analizy danych geodezyjnych, była przedmiotem ostatnich prac teoretycznych i praktycznych zespołu (patrz rozdz. 6).

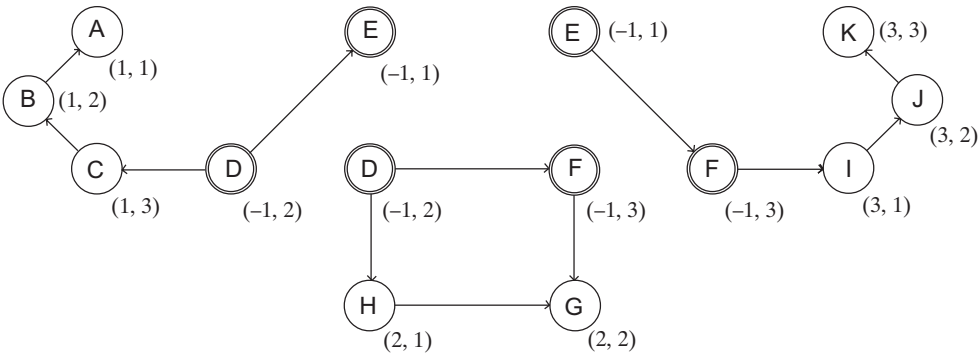
2. Równoległe transformacje grafowe

Jednym z podstawowych mankamentów transformacji grafowych jest ich niska efektywność w przetwarzaniu dużych grafów. Rozwiązaniem tego problemu jest zrównoleglenie wykonywania transformacji grafu. Podstawowym założeniem było wprowadzenie mechanizmu, który pozwoli na uniknięcie dodatkowego nakładu pracy projektanta systemu związanego z wprowadzeniem synchronizacji w trakcie równoległego wykonywania transformacji.

Koncepcja podziału grafu scentralizowanego na podgrafy w taki sposób, by po ich lokalnej modyfikacji można było je skleić w jeden spójny scentralizowany graf, została po raz pierwszy przedstawiona w publikacji Kotulskiego (2008) dla gramatyk typu algo-

rytmicznego (NLC) i była rozwijana również w odniesieniu do podejścia algebraicznego (w kontekście DPO (Kotulski i Sędziwy 2009) i w kontekście SPO (Kotulski i Sedziwy 2009)) oraz dla notacji hipergrafowej (Kotulski i Strug 2009).

Generalnie zakładamy, że graf scentralizowany podzielony jest na kilka grafów zwanych komplementarnymi, przy czym podział przebiega wierzchołkami, tzn. wierzchołki graniczne zostają zreplikowane i umieszczone w odpowiednich grafach komplementarnych. Graficznie zreplikowane wierzchołki graniczne są oznaczone jako podwójne okręgi (patrz rys. 1). Zatem wierzchołek etykietowany jako D ma swoją reprezentację w G_1 i G_2, E w G_1 i G_3, a F w G_2 i G_3.



Rys. 1. Grafy komplementarne $\{G_1, G_2, G_3\}$ powstałe z podziału G
Źródło: Kotulski (2000)

Synchronizacja opiera się na następujących założeniach:

- każdym grafem komplementarnych opiekuje się jeden agent obliczeniowy, który sekwencyjnie wykonuje transformacje;
- transformacja może być wykonana tylko na danych lokalnych grafu komplementarnego (tzn. wyłączone są z niej wierzchołki graniczne);
- w momencie gdy produkcja chce wykorzystać wierzchołek graniczny, agenci współdzielący go negocjują przesunięcie granicy tak, by stał się on wierzchołkiem lokalnym (co umożliwia wykonanie kroku poprzedniego).

Ze względów efektywnościowych koncepcja zreplikowanych grafów komplementarnych RCG (*replicated complementary graph*) dopuszcza replikowanie pomiędzy grafami również elementów niebędących wierzchołkami granicznymi (Kotulski 2000). Koncepcja ta jest wspierana przez wieloagentowe środowisko GRADIS, w którym zakładamy, że każdy agent kontroluje pojedynczy graf komplementarny G_i.

Niskopoziomowy mechanizm wymiany komunikatów i podsystem synchronizacji agentów jest ukryty przed użytkownikiem, który operuje na poziomie abstrakcji zdefiniowanym przez następujące operacje pakietu GRADIS: SplitGraph, JoinGraphs,

Incorporate, Neighborhood dla klasycznej koncepcji grafów komplementarnych oraz Expand i RemoveRepl dla wersji z replikacjami.

Pierwsza z tych operacji wspiera kreowanie agentów i podział informacji w postaci grafów komplementarnych pomiędzy tymi agentami. Operacja SplitGraph (G, V_H) dla danego grafu G_{01} i podzbioru wierzchołków $V_H \subset V(G_{01})$ produkuje dwa grafy komplementarne G_{10} i G_{11} w taki sposób, że G_{11} zawiera wierzchołki należące do V_H, a G_{10} pozostałe wierzchołki grafu G_{01} oraz zreplikowane wierzchołki graniczne. Aby podzielić graf G_{01} zgodnie ze zbiorami $\{V_i\}_{i=1,...,p}$ ($V_i \cap V_j = \emptyset$ dla $i \neq j$ i $\bigcup_{i=1}^{p} = V(G_{01})$), agent A powinien iteracyjnie wykonywać operację SplitGraph (G_{k1}, V_{k+1}), gdzie dla G_{kj} k oznacza numer iteracji ($k = 0, 1, ..., p - 1$), a $j = 0, 1$ są odpowiednio indeksami podgrafów wygenerowanych w danej iteracji.

Udowodniono (Kotulski 2000), że w przypadku grafów RCG wywiedzionych w ustalonej gramatyce $\mathcal{G}$ podział dowolnego z nich $G_i \in \mathcal{G}$ na dwa inne G_i', G_i'' zachowuje własności tej gramatyki $G' \in \mathcal{G} \wedge G'' \in \mathcal{G}$ oraz G_i' i G_i'' są wzajemnie komplementarne. Ponadto operacja SplitGraph nie wpływa na inne grafy z wyjątkiem dzielonego G_i (tylko wewnętrzne dane w replikowanych wierzchołkach, utrzymywane w celu poprawy efektywności komunikacji agentów, mogą być zmienione).

Operacja JoinGraphs() rozsyła do wszystkich agentów żądanie przesłania kopii grafów komplementarnych utrzymywanych przez nie; następnie skleja je (w trybie transakcyjnym) i w rezultacie odzyskuje scentralizowany graf G.

Operacja Incorporate(v, i) przesuwa granice pomiędzy grafami komplementarnymi.

W przypadku transformacji grafowych pierwszą operacją jest znalezienie mapowania m grafu lewej strony produkcji L w grafie G_i. Możemy tu rozważyć dwie sytuacje:

1) $m(L) \subseteq G_i$ – wtedy wiemy, że możemy zastosować tę produkcję w kontekście grafu G_i, chociaż w niektórych typach gramatyk, itp. SPO, do realizacji transformacji będzie wymagane założenie, by $V_L \subseteq$ Private(G_i);
2) tylko fragment $m(L)$ zawiera się w G_i – wtedy będziemy musieli odtworzyć otoczenie pewnego wierzchołka $v \in m(L) \cap G_i$, bazując na informacjach z innych grafów komplementarnych; bezpośrednie przesuwanie granicy pomiędzy wierzchołkami byłoby nieefektywne, wprowadzono więc operację $O =$ neighborhood(v, G, k) (gdzie k jest maksymalną odległością od v wierzchołków należących do tego otoczenia).

Pomimo wielomianowej złożoności obliczeniowej operacja Neighborhood() realizująca protokół 2-commit jest operacją złożoną i czasochłonną.

Alternatywną opcją jest stworzenie mechanizmów umożliwiających obserwowalność struktur innych podgrafów komplemetarynych przez wprowadzenie możliwości replikacji innych wierzchołków niż brzegowe.

Operacja Expand(v, i, COND) dla każdej repliki u danego wierzchołka v grafu G_i tworzy replikę sąsiadów replik u, którzy spełniają warunek COND; nowe repliki są dołączane do G_i razem z krawędziami łączącymi je z u. W ten sposób jesteśmy w stanie zbudować szersze niż brzegowe część wspólne widoczne z dwóch grafów komplementarnych.

Operacja RemoveRepl(v, i) jest operacją odwrotną, czyli usuwa repliki niebędące wierzchołkami brzegowymi wraz łączącymi je krawędziami komplementarnymi.

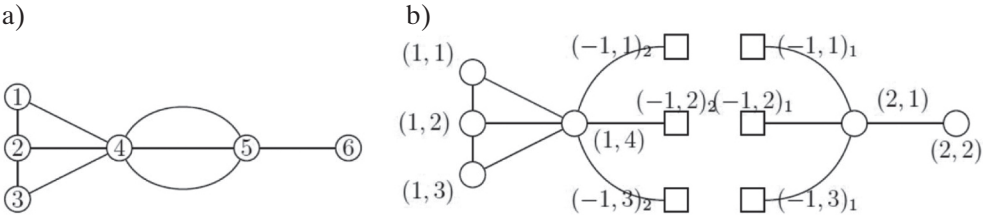
Podstawy formalne wykorzystania koncepcji RCG wyrażają się w następujących – udowodnionych przez Kotulskiego (2000) – stwierdzeniach:

- Wprowadzony mechanizm zastosowania transformacji grafowych zgodnie z koncepcją grafów komplementarnych:
  - zachowuje własności gramatyki $\mathcal{G}$ zbadane w kontekście grafów scentralizowanych; w szczególności twierdzenia Churcha–Rossera i o równoległości, prawdziwe w środowisku scentralizowanym, będą też prawdziwe w środowisku rozproszonym;
  - nie powoduje zakleszczenia.
- Operacje wspomagające koncepcję mają złożoność wielomianową.

Koncepcja grafów ciętych (*slashed graphs*) jako grafowej reprezentacji rozproszonej, uzyskanej z grafu centralnego G, wprowadzona przez Adama Sędziwego (Sędziwy 2012), bazująca na idei grafów komplementarnych, redukuje liczbę powiązań pomiędzy agentami utrzymującymi grafy częściowe. W odniesieniu do grafów ciętych przyjęto założenie, że zamiast replikowania wierzchołków rozcinamy krawędzie łączące grafy częściowe.

Na rysunku 2 przedstawiono przykładowy graf G (rys. 2a) oraz jego postać ciętą (rys. 2b), w której wierzchołki główne oznaczono jako °, a wierzchołki atrapowe za pomocą □. Wprowadza się następującą konwencję indeksacyjną dla wierzchołków grafu ciętego (zob. rys. 2). Wierzchołek główny indeksowany jest parą (i, k), gdzie i jest unikatowym w $\mathcal{G}$ identyfikatorem grafu ciętego G_i, a k jest unikatowym w G_i indeksem tego wierzchołka. Wierzchołek atrapowy indeksowany jest parą $(-1, k)_r$, gdzie k jest globalnie unikatowym identyfikatorem wierzchołka (a ściślej: wierzchołka i jego kopii), a r jest referencją do grafu ciętego zawierającego replikę tego węzła. Wprowadzenie indeksu r umożliwia natychmiastową lokalizację repliki węzła atrapowego. Zgodnie z powyższą konwencją dla replik węzła atrapowego v i v' mamy: $v = (-1, k)_{r_1} \in V(G_{r_2})$, $v' = (-1, k)_{r_2} \in V(G_{r_1})$.

Zysk z powyższego rozwiązania wynika z tego, że w przypadku grafu krawędź może być replikowana jedynie pomiędzy dwa grafy komplementarne (w RCG wierzchołek mógł być replikowany do kilku grafów), co upraszcza protokoły komunikacyjne. Przewaga ta nie występuje w przypadku hipergrafów oraz rozwiązań wykorzystujących dodatkową replikację wierzchołków.



Rys. 2. Graf G (a), reprezentacja $\mathcal{G}$ (b)
Źródło: Sędziwy (2012)

4. Wykorzystanie transformacji grafowych w projektowaniu architektonicznym

Wspomaganie projektowania systemów architektonicznych było jednym z podstawowych, wymienionych w monografii pod redakcją Ehriga i in. (1999a), zastosowań metod formalnych omawianych przez Engelfrieta i Rozenberga (1997).

Większość z tych badań bazuje na scentralizowanej reprezentacji grafowej, co – z jednej strony – z powodu niskiej efektywności obliczeniowej ogranicza ich zastosowanie do grafów małych rozmiarów. Z drugiej jednak strony w trakcie wspomagania projektowania wiele z problemów architektonicznych można zdekomponować na podproblemy i analizować je równolegle. Problemem tym zajęto się we współpracy z zespołem z Uniwersytetu Jagiellońskiego.

Dekompozycja ta może mieć charakter horyzontalny i wertykalny. Przykładem pierwszego typu dekompozycji jest projektowanie bloków czy hoteli, gdzie każde mieszkanie/apartament można projektować oddzielnie. Tutaj automatycznie i efektywnie można zastosować koncepcję grafów komplementarnych. Jedynym wyzwaniem było dobranie odpowiedniej reprezentacji problemu. Początkowo rozważane były grafy (Kotulski i Strug 2009), później jednak bardziej intuicyjne wydawały się rozwiązania oparte na hipergrafach (Kotulski i Strug 2011, 2012).

Przykładem dekompozycji wertykalnej jest wykorzystywanie różnych gramatyk grafowych do różnych podproblemów oraz synchronizowanie uzyskanych lokalnie efektów itp.:

- projektowanie struktury pomieszczeń (ścian, drzwi, okien itp.) oraz projektowanie wyposażenia,
- synchronizacja równoległego projektowania elewacji budynku oraz struktury wewnętrznej.

W przypadku pierwszym kooperacja gramatyk jest prostsza, gdyż gramatyka wspomagająca projektowanie wyposażenia jest zsynchonizowana lokalnie z wierzchołkiem grafu struktury reprezentującym pewien obszar powierzchni.

W drugim przypadku synchronizacja dwóch gramatyk jest trudniejsza, gdyż pewne działania mogą się wykluczać. Przykładowo nie możemy przesunąć okna w elewacji budynku bez wykonania odpowiednich akcji w strukturze wewnętrznej budynku. Od strony badawczej wymagało to rozwiązania kilku problemów. Pierwszy z nich jest związany ze sterowaniem wywodem w gramatyce. We wcześniejszych rozwiązaniach prosty automat skończenie stanowy (zwany diagramem sterującym) był wystarczający do zapewnienia kontroli wykonania produkcji (Kotulski 2000). W kontekście sprzęgania kilku gramatyk bardziej złożony mechanizm był konieczny. W publikacji Kotulskiego i in. (2014) zdefiniowano język sterowania, w którym kluczowym elementem jest możliwość transakcyjnego wykonania sekwencji produkcji, czyli zapewnienia sobie, że albo wszystkie produkcje w sekwencji się wykonają, albo nie pozostanie efekt wykonania żadnej z nich. Sprzężenie pomiędzy gramatykami jest też oparte na mechanizmie transakcji, co oznacza, że wykonanie produkcji w jednej gramatyce jest zależne od poprawnego wykonania produkcji (lub ich sekwencji) w drugiej gramatyce. Transakcyjność na poziomie sprzężenia była jednak możliwa, dopiero gdy wprowadzono transakcyjność na poziomie każdej z gramatyk z osobna (w ramach języka kontroli wykonania produkcji). Od 2014 roku prace dotyczące programowania architektonicznego są prowadzone wyłącznie w zespole badawczym na Uniwersytecie Jagiellońskim z uwagi na podjęcie przez zespół GRADSI badań w nowym bardzo obiecującym obszarze praktycznych zastosowań gramatyk grafowych.

4 Gramatyki dualne

Kolejnym obszarem zastosowań gramatyk grafowych okazało się projektowanie instalacji oświetleniowych. Pierwszym tematem badawczym była kontrola dynamicznego sterowania oświetleniem na stacji benzynowej rozważana w publikacji Wojnickiego i in. (2014), w której zdefiniowano koncepcję CAG (*control avaiability graph*). Równolegle zdiagnozowano istotne praktyczne zapotrzebowanie na wspomaganie projektowania oświetlenia. Normy oświetleniowe, przykładowo EN 13201-2:2015, precyzyjnie definiują wymagania oświetleniowe; na ich spełnienie ma wpływ kilkadziesiąt takich parametrów, jak: wysokość i odstęp słupów, na których lampy są zawieszone, długość wysięgnika i kąt zawieszenia, klasa oświetleniowa ulicy i jej szerokość oraz szereg innych. Jeśli uwzględni się ponadto dopuszczalne stopnie swobody dyskretyzacji tych parametrów, daje to ponad $2{,}7 \cdot 10^{22}$ wariantów doboru parametrów. Sprawdzenie ich wszystkich za pomocą komputera w rozsądnym czasie jest niemożliwe. Człowiek (ekspert) dobiera je intuicyjnie (i dalej system komputerowy weryfikuje poprawność tego doboru), ale nie są to wybory optymalne.

Zbudowanie gramatyki grafowej wspomagającej proces projektowania i analizowanie wyłącznie wariantów parametrów wyznaczonych przez tę gramatykę (formalnie będących elementami języka zbudowanego nad tą gramatyką) ograniczyło ich liczbę do kilkunastu milionów. Taką liczbę komputer już jest w stanie przeanalizować (chociaż problem efektywności tego przetwarzanie pozostaje w dalszym ciągu aktualny i będzie jeszcze rozważany w rozdziale 5). Efektywność eksperckich projektów wymiany oświetlenia sodowego na oświetlenie LED wynosi około 50% (tak szacuje NFOŚiGW), podczas gdy Phoca generuje projekty o efektywności powyżej 65%. Wyższa efektywność wymiany przekłada się nie tylko na oszczędność energii elektrycznej, ale również ogranicza negatywny wpływ prześwietlenia nocnego (*light pollution*) na zdrowie człowieka.

Różne aspekty tworzenia gramatyki wspomagającej projektowanie były przedmiotem licznych publikacji, w szczególności w czasopismach z listy filadelfijskiej (Sędziwy 2016, Sędziwy i Kotulski 2016a, 2016b, Wojnicki i in. 2016a, 2016b, 2019, Bielecka i in. 2017, Kotulski i in. 2017, Sędziwy i Basiura 2018, Wojnicki i Kotulski 2018a, Ernst i in. 2018, 2020, 2021, Peña-García i Sędziwy 2020, Basiura i in. 2021, Ernst i Kotulski 2021).

Normy oświetleniowe dopuszczają obniżenie poziomu oświetlenia (zmianę klasy oświetleniowej), gdy zmienią się warunki zewnętrzne. Zwykle dotyczy to zmiany natężenia ruchu, ale może też dotyczyć charakterystyki ulicy (np. od określonej godziny dopuszczone lub zabronione jest parkowanie przy niej). Pod względem technologicznym lampy LED umożliwiają dynamiczną modyfikację poziomu oświetlenia (*dimming*), przy czym poziom nowego oświetlenia jest wynikiem nowego projektu. Schemat, w jaki zmiana parametrów wejściowych (danych z sensorów) wpływa na poziom oświetlenia, opisywany jest w CAG (Wojnicki i in. 2014, 2017) i był praktycznie wykorzystany do realizacji dynamicznego oświetlenia w Krakowie. CAG był jednak generowany ręcznie dla konkretnych sytuacji oświetleniowych.

Automatyczna generacja CAG wymaga:

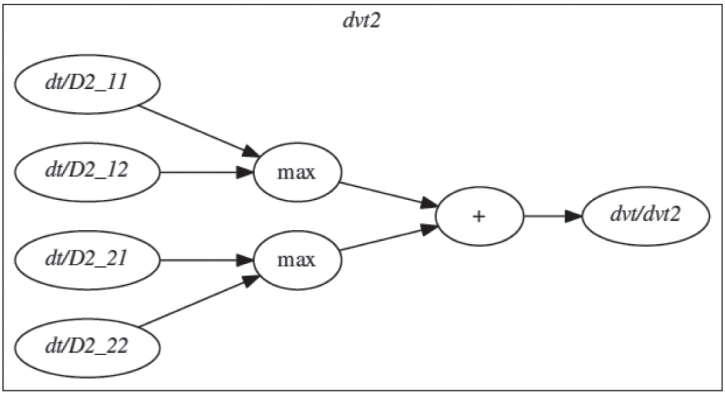
- analizy warstwy sensorycznej zbierającej informację, tworzącej DSG (*detector structure graph*),
- analizy i zaprojektowania warstwy oświetleniowej tworzącej CAG.

Kluczowym pojęciem występującym w obu strukturach jest wierzchołek reprezentujący „wirtualny sensor” (rys. 3, 4). Gramatyka grafowa θ tworząca DSG, czyli wyliczająca wartości sensorów wirtualnych na podstawie danych z sensorów, przypomina gramatyki generujące wyrażenia arytmetyczne. Pozwala też na łatwe i niezależne od CAG modyfikacje struktury sensorycznej w razie jej rozbudowy czy awarii. Gramatyka Ψ kontrolująca proces generacji grafu CAG wykorzystuje wierzchołki „sensorów

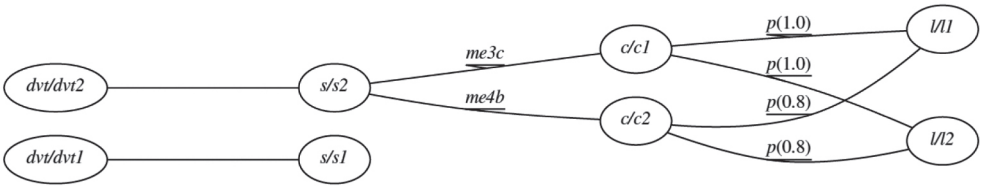
wirtualnych" do wyboru odpowiedniego wariantu oświetlenia (a w konsekwencji poziomu oświetlenia poszczególnych lamp). Gramatyką dualną (Wojnicki i Kotulski 2018b) nazywamy parę gramatyk [Ψ, Θ] wraz z regułami synchronizacji ich pracy: gdy zmieniana jest struktura wspólna dla grafów generowanych przez obie gramatyki, zmieniają się wartości atrybutów tej wspólnej struktury grafowej.

W rezultacie wprowadzenia gramatyk dualnych wzrasta czytelność modelu formalnego dzięki odseparowaniu różnych punktów widzenia. Ciekawą informacją jest to, że docelowo liczba krawędzi w grafie DSG + CAG jest większa niż liczba krawędzi w grafach generowanych pod kontrolą gramatyk dualnych. W przypadku rozwiązania będącego implementacją dynamicznego sterowania w Krakowie ten stosunek wynosił 2,8.

W praktycznych rozwiązaniach bardzo łatwo uogólnić tę koncepcję na gramatyki parami dualne.



Rys. 3. Gramatyka dualna – DSG



Rys. 4. Gramatyka dualna – CAG

5. Opóźnione wykonywanie produkcji

Generowanie akceptowalnych wariantów parametrów oświetleniowych przez gramatyki grafowe odgraniczyło ich liczbę do kilku milionów wariantów, jednak przy opracowywaniu projektów dla dużych miast (oświetlanych przez kilkadziesiąt tysięcy lamp)

niezwykle istotnym czynnikiem jest przyśpieszenie obliczeń. Do działań umożli-wiających to należy z pewnością zrównoleglanie obliczeń (zgodnie z koncepcją *complementary/slashed graphs*) oraz upraszczanie (zmniejszanie rozmiaru) analizowanego grafu wynikowego (*dual graph grammars*).

Analizując proces optymalnego projektowania systemu oświetleniowego zgodnie z normą EN-PN 13201, można zauważyć podobieństwo niektórych sytuacji obliczeniowych. Celowa byłaby jednoczesna optymalizacja tych sytuacji, przy czym oczekuje się automatyzacji tych działań przez system grafowy. W tym celu zaproponowany został (Sędziwy i Basiura 2018, Basiura 2019, Basiura i in. 2021) model formalny bazujący na dwóch typach grafów. Pierwszy z nich, graf IAG (*infrastructure abstract graph*), wykorzystywany jest do modelowania statycznych informacji na temat badanych obiektów, z kolei graf KAG (*knowledge abstract graph*) służy modelowaniu reguł projektowych i bazy wiedzy.

Całość modelu operuje na transformacjach grafowych pozwalających na redukcję rozmiaru przestrzeni poszukiwań. Domknięciem zaproponowanego modelu formalnego jest efektywna metoda przetwarzania danych. W pracy Basiury (2019) wprowadzono tzw. mechanizm „opóźnionej produkcji". Umożliwia on optymalizację całego procesu w efekcie jego ciągłej analizy i wykorzystania cząstkowych wyników oraz redukcji rozmiaru przestrzeni poszukiwań do problemów przetwarzanych w dalszej kolejności. Chwilowe „opóźnienie" umożliwia też wykorzystanie wyników operacji na grafach „podobnych" wykonywanych równolegle w systemie wieloagentowym. Zastosowanie tej strategii umożliwia przyspieszenie całego procesu przetwarzania grafów przy zapewnieniu pełnej skalowalności. Testy praktyczne metody w środowisku złożonym z pięciu równoległe pracujących serwerów dla obszaru testowego w Krakowie (3768 lamp) wykazały 16-krotne przyspieszenie obliczeń i pozwoliły na wykonanie 606 projektów w 3 godz. 52 min.

6. Gramatyki grafowe wyzwalane przez analizę danych

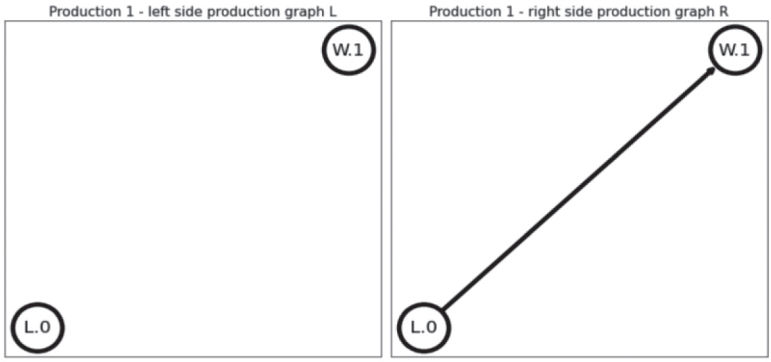
Automatyzacja procesu projektowania oświetlenia oprócz analizy struktury grafu modelującego problem wymaga analizy specyficznych atrybutów jego składników, takich jak kształt dróg lub lokalizacja słupów oświetleniowych. Dane te, stanowiące komponent przestrzenny, komplementarny względem komponentu atrybutowego, pozyskiwane są zazwyczaj z systemów typu GIS (*Geographic Information Systems*), takich jak np. OpenStreetMap[1].

[1] https://www.openstreetmap.org.

Analiza związków przestrzennych (a więc np. odległości, przecinania się, równoległości) oraz kształtów (np. odróżnienie prostych odcinków drogi od zakrętów) wymaga zastosowania nietrywialnych procedur, które zazwyczaj implementowane są przez odpowiednie biblioteki lub rozwiązania bazodanowe.

Jednocześnie mamy świadomość, że docelowo przetworzone/pozyskane informacje mają stworzyć strukturę formalną umożliwiającą wnioskowanie, analizę i optymalne projektowanie. Zaproponowany w rozdziale 3 język sterowania wywodem okazał się zbyt ograniczony w tym zakresie.

Rozważmy następujący przykład. Przy założeniu, że mamy zdefiniowaną sieć drogową (wierzchołki typu *way*) i zbiór lamp ulicznych (wierzchołki typu *lamp*), chcielibyśmy określić, która lampa może oświetlać daną drogę (krawędź *influences*). Zarówno lampa, jak i droga mają określone swoje położenie (droga jako odcinek), w dużym uproszczeniu (nie analizując przeszkód terenowych) możemy więc przyjąć, że lampa wpływa na oświetlenie ulicy, gdy odległość lampy od ulicy jest mniejsza od zadanej wartości granicznej, np. 50 metrów. W przypadku zastosowania diagramów sterujących wywodem realizacja tego działania oparta byłaby na podejściu kombinatorycznym, dla każdej pary lamp (*P*, *W*), gdy ich odległość jest mniejsza niż 30 metrów, uruchamiamy produkcję wskazaną na rysunku 5. Ponieważ liczbę odcinków ulic i lamp liczy się w tysiącach, ten sposób realizacji zadania nie jest efektywny.



Rys. 5. Produkcja dodająca relację

Przy uwzględnieniu relacji geodezyjnych wyznaczenie par (*P*, *W*) spełniających kryterium odległości jest standardową operacją, a więc możemy zrezygnować z kontroli wykonywania produkcji za pomocą diagramów sterujących. Zamiast analizować liczbę sytuacji będących iloczynem liczby lamp i odcinków dróg, analizujemy tylko przypisanie lampy do wskazanego odcinka. W podanym przykładzie relacja geoprzestrzenna

jest bardzo prosta – odległość między dwoma obiektami, w rzeczywistości jednak mamy do czynienia z bardziej skomplikowanymi sytuacjami typu:

- wpływ przesłaniania obiektów,
- wpływ ich kształtu,
- wpływ relacji położenia i przecinania.

Stąd kluczowe okazało się opracowanie i wdrożenie metodyki wyzwalania transformacji grafowych (oraz określania wartości atrybutów wierzchołków/krawędzi biorących udział w transformacji) z systemem wnioskowania na bazie informacji geoprzestrzennych. Ten mechanizm rozwijany jest pod nazwą *Graph Grammar Triggered by Spatial Relations*.

7. Przykłady zastosowań

Prace teoretyczne weryfikowane były w praktyce. Pierwszy duży projekt realizowano w ramach grantu ISE (finansowanego przez NFOŚiGW i Miasto Kraków) – w dzielnicy Kraków-Krowodrza. W projekcie tym zweryfikowano zarówno koncepcję automatycznego wspomagania projektowania, jak i dynamicznego sterowania. W jego wyniku osiągnięto:

- poprawę efektywności energetycznej; wymiana oświetlenia według projektu wspomaganego przez rozwiązanie Phoca zmniejszyła zużycia energii na poziomie 71% (z tego 15% dzięki dynamicznemu sterowaniu); dla porównania w projektach przygotowywanych przez projektanta wspomaganego narzędziami weryfikującymi typu Dialux (branżowe oprogramowanie do weryfikacji projektów oświetleniowych) zysk ten wynosił jedynie 40%;
- skrócenie czasu projektowania do kilku godzin w porównaniu z siedmioma tygodniami pracy zespołu projektantów.

Obszar projektu Kraków – ISE stał się poligonem do testowania nowych rozwiązań dotyczących:

- przyspieszenia obliczeń (w tym ich zrównoleglenia),
- uproszczenia modelu formalnego,
- wpływu nowych rozwiązań oświetleniowych na wzrost efektywności energetycznej systemu oświetleniowego (wzrost efektywności o około 10%),
- analiz wpływu nowych norm na efektywność (zawarte w normie EN-PN 13201 z 2014 roku rozwiązania umożliwiają wzrost efektywności dynamicznego sterowania o 40,8% w porównaniu z normą z 2007 (Wojnicki i in. 2019).

O skalowalności rozwiązania świadczą projekty oświetleniowe dla Tbilisi (96 tys. punktów świetlnych) oraz Waszyngtonu (54 tys. punktów świetlnych), gdzie obliczenia

trwały zaledwie kilkanaście godzin. Ma to istotne znaczenie, gdy chcemy przeanalizować kilka wariantów oświetlenia, np. różnych producentów lamp. System sprawdził się też w produkcji, o czym świadczy wykonanie kilkuset projektów dla miast, gmin oraz operatorów energetycznych.

Dzięki obliczeniom w systemie Phoca możliwe stało się wyznaczenie przedziałów kosztu pozyskania 1 MWh energii (LCOE – *levelized cost of elecricity*). W przypadku wymiany oświetlenia koszt redukcji zużycia energii o 1 MWh (traktowany jako koszt pozyskania) wynosi od $LCOE_{retrofit}$ do [23,06 EUR, 54,64 EUR]; minimalna wartość to projekt optymalny, podczas gdy wartość maksymalna określa minimum efektywności przyjmowane w projektach infrastrukturalnych (obecnie 50%) (Kotulski i in. 2021). Taka efektywność czyni wymianę oświetlenia najkorzystniejszą formą pozyskania zielonej energii, tym korzystniejszą, im bardziej efektywny będzie projekt.

Dla porównania Fraunhofer Institute (Kost i in. 2018) obliczył LCOE dla następujących źródeł energii:

- lądowe turbiny wiatrowe – od 39,9 EUR/MWh do 82,3 EUR/MWh,
- systemy fotowoltaiczne (PV) – od 37,1 EUR/MWh do 115,4 EUR/MWh,
- morskie turbiny wiatrowe – od 74,9 EUR/MWh do 137,9 EUR/MWh,
- biogaz – od 101,4 EUR/MWh do 147,4 EUR/MWh,
- konwencjonalne elektrownie:
  - węgiel brunatny – od 45,9 EUR/MWh do 79,8 EUR/MWh,
  - węgiel kamienny – od 62,7 EUR/MWh do 98,6 EUR/MWh.

8. Podsumowanie

Od kilkunastu lat grupa badawcza GRADIS zajmuje się rozwojem specyfikacji systemów za pomocą transformacji grafowych zarówno od strony czysto teoretycznej (zastosowania), jak i efektywnościowej (w kontekście zrównoleglania obliczeń). Efektem prac teoretycznych jest kilkadziesiąt publikacji członków zespołu, w ostatnim okresie w większości za minimum 80 pkt w punktacji MEiN.

Istotne jest również, że efekty tych badań znalazły zastosowanie w praktyce, czego dowodem jest zgłoszenie rozwiązania jako osiągnięcia komercjalizacji wyników w dyscyplinie informatyka techniczna i telekomunikacja w ramach ewaluacji uczelni.

Również Polski Komitet Normalizacji zgłosił rozwiązanie GRADIS do nagrody CEN-CENELEC Standards + Innovation Awards 2021, gdzie znaleźliśmy się w ścisłym finale w gronie dziewięciu innych projektów europejskich:

Widoczny jest rozwój liczebny i jakościowy zespołu. Początkowo był to zespół dwuosobowy, z jednym samodzielnym pracownikiem naukowym. Obecnie mamy już

czterech samodzielnych pracowników naukowych, kilku doktorów oraz wiele osób współpracujących z zespołem. To pozwala optymistycznie patrzyć w przyszłość, szczególności w aspekcie powiązania technik opartych na transformacjach grafowych oraz uczeniu maszynowym.

Literatura

Basiura A., 2019, *Grafowy model formalny współbieżnego projektowania oświetlenia ulicznego*, Akademia Górniczo Hutnicza, Kraków [rozprawa doktorska].

Basiura A., Sędziwy A., Komnata K., 2021, *Similarity and conformity graphs in lighting optimization and assessment*, [w:] Paszynski M., Kranzlmüller D., Krzhizhanovskaya V.V., Dongarra J.J., Sloot P.M. (eds.), *Computational Science – ICCS 2021: 21st International Conference: Krakow, Poland, June 16–18, 2021: proceedings, Pt. 4*, Lecture Notes in Computer Science, vol. 1274, Springer, Cham, s. 145–157. https://doi.org/10.1007/978-3-030-77970-2_12.

Bielecka M., Bielecki A., Ernst S., Wojnicki I., 2017, *Prediction of traffic intensity for dynamic street lighting*, [w:] Ganzha M., Maciaszek L., Paprzycki M. (eds.), *FedCSIS: Proceedings of the 2017 Federated Conference on Computer Science and Information Systems: September, 3–6, 2017, Prague, Czech Republic*, PTI, IEEE, Piscataway, s. 1165–1171. https://doi.org/10.15439/2017F389.

Ehrig H., Engels G., Kreowski H.-J., Rozenberg G. (eds.), 1999a, *Handbook of Graph Grammars and Computing by Graph Transformation. Volume 2: Applications, Languages and Tools*, World Scientific, Singapore.

Ehrig H., Kreowski H.-J., Montanari U., Rozenberg G. (eds.), 1999b, *Handbook on Graph Grammars and Computing by Graph Transformation. Volume 3: Concurrency, Parallelism, and Distribution*, World Scientific, Singapore.

Ehrig H., Ermel C., Golas U., Hermann F., 2015, *Graph and Model Transformation*, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-47980-3.

EN 13201-2:2015, *Road Lighting – Part 2: Performance requirements*, European Committee for Standardization, Brussels.

Engelfriet J., Rozenberg G., 1997, *Node replacement graph grammars*, [w:] Rozenberg G. (ed.), *Handbook of Graph Grammars and Computing by Graph Transformation. Volume 1: Foundations*, World Scientific, Singapore, s. 1–94. https://doi.org/10.1142/9789812384720_0001.

Ernst S., Kotulski L., 2021, *Estimation of road lighting power efficiency using graph-controlled spatial data interpretation*, [w:] Paszynski M., Kranzlmüller D., Krzhizhanovskaya V.V., Dongarra J.J., Sloot P.M.A. (eds.), *Computational Science – ICCS 2021: 21st International Conference: Krakow, Poland, June 16–18, 2021: proceedings, Pt. 1*, Lecture Notes in Computer Science, vol. 12742, Springer, Cham, s. 585–598. https://doi.org/10.1007/978-3-030-77961-0_47.

Ernst S., Łabuz M., Środa K., Kotulski L., 2018, *Graph-based spatial data processing and analysis for more efficient road lighting design*, Sustainability, vol. 10, 3850. https://doi.org/10.3390/su10113850.

Ernst S., Kotulski L., Lerch T., Rad M., Sędziwy A., Wojnicki I., 2020, *Calculating reactive power compensation for large-scale street lighting*, [w:] Krzhizhanovskaya V.V., Závodszky G., Lees M.H., Dongarra J.J., Sloot P.M.A., Brissos S., Teixeira J. (eds.), *Computational Science – ICCS 2020: 20th International Conference: Amsterdam, The Netherlands, June 3–5, 2020: proceedings, Pt. 2*, Lecture Notes in Computer Science, vol. 12138, Springer, Cham, s. 538–550. https://doi.org/10.1007/978-3-030-50417-5_40.

Ernst S., Kotulski L., Rad M., Sędziwy A., Wojnicki I., 2021, *Application of reactive power compensation algorithm for large-scale street lighting*, Journal of Computational Science, vol. 51, 101338. https://doi.org/10.1016/j.jocs.2021.101338.

Kost C., Shammugam S., Jülch V., Nguyen H.-T., Schlegl T., 2018, *Levelized Cost of Electricity: Renewable Energy Technologies*, Fraunhofer Institute for Solar Energy Systems ISE. https://www.ise.fraunhofer.de/content/dam/ise/en/documents/publications/studies/EN2018_Fraunhofer-ISE_LCOE_Renewable_Energy_Technologies.pdf, 2018.

Kotulski L., 2000, *Model wspomagania generacji oprogramowania współbieżnego w środowisku rozproszonym za pomocą gramatyk grafowych*, Rozprawy Habilitacyjne Uniwersytetu Jagiellońskiego, nr 349, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków.

Kotulski L., 2008, *Distributed graphs transformed by multiagent system*, [w:] Rutkowski L., Tadeusiewicz R., Zadeh L.A., Zurada J.M. (eds.), *Artificial Intelligence and Soft Computing – ICAISC 2008: 9th International Conference: Zakopane, Poland, June 22–26, 2008: proceedings*, Lecture Notes in Computer Science, vol. 5097, Springer, Berlin, Heidelberg, s. 1234–1242.

Kotulski L., Sedziwy A., 2009, *On complexity of coordination of parallel graph transformations in GRADIS framework*, [w:] Zamojski W., Mazurkiewicz J., Sugier J., Walkowiak T. (eds.), *DepCoS – RELCOMEX '09: Proceedings of International conference on Dependability of Computer Systems: Brunów, Poland, 30 June–2 July, 2009*, IEEE, Piscataway, s. 279–289. https://doi.org/10.1109/DepCoS-RELCOMEX.2009.51.

Kotulski L., Sędziwy A., 2009, *Zastosowanie gramatyk grafowych typu double pushout w środowisku GRADIS*, Automatyka, t. 13, z. 3, s. 911–920.

Kotulski L., Strug B., 2009, *Using graph transformations in distributed adaptive design system*, [w:] Bolc L., Kulikowski J.L., Wojciechowski K. (eds.), *Computer Vision and Graphics: International Conference, ICCVG 2008, Warsaw, Poland, November 10–12, 2008: revised papers*, Lecture Notes in Computer Science, vol. 5337, Springer, Berlin, Heidelberg, s. 477–486. https://doi.org/10.1007/978-3-642-02345-3_47.

Kotulski L., Strug B., 2011, *Multi-agent system for distributed adaptive design*, Key Engineering Materials, vol. 486, s. 217–220. https://doi.org/10.4028/www.scientific.net/KEM.486.217.

Kotulski L., Strug B., 2012, *Supporting communication and cooperation in distributed representation for adaptive design*, Advanced Engineering Informatics, vol. 27(2), s. 220–229. https://doi.org/10.1016/j.aei.2012.10.002

Kotulski L., Sedziwy A., Strug B., 2014. *Heterogeneous graph grammars synchronization in CAD systems supported by hypergraph representations of buildings*, Expert Systems with Applications, vol. 41(4), pt. 1, s. 990–998. https://doi.org/10.1016/j.eswa.2013.07.043.

Kotulski L., Sędziwy A., Strug B., Wojnicki I., Ernst S., 2017, *Synchronisation methods in graph-based knowledge representation for large-scale design process*, International Journal of Design Engineering, vol. 7, s. 17–32. https://doi.org/10.1504/IJDE.2017.085640.

Kotulski L., Basiura A., Wojnicki I., Siuchta S., 2021, *Lighting system modernization as a source of green energy*, Energies, vol. 14(10), 2771 https://doi.org/10.3390/en14102771.

Peña-García A., Sędziwy A., 2020, *Optimizing lighting of rural roads and protected areas with white light: a compromise among light pollution, energy savings, and visibility*, LEUKOS, vol. 16, no. 2, s. 147–156. https://doi.org/10.1080/15502724.2019.1574138.

Rozenberg G. (ed.), 1997, *Handbook of Graph Grammars and Computing by Graph Transformation. Volume 1: Foundations*, World Scientific, Singapore.

Sędziwy A., 2012, *Effective graph representation for agent-based distributed computing*, [w:] Jezic G., Kusek M., Nguyen N.T., Howlett R.J., Jain L.C. (eds.), *Agent and Multi-agent Systems: Technologies and Applications: 6th KES International Conference, KES-AMSTA 2012: Dubrovnik, Croatia, June 25–27, 2012: proceedings*, Lecture Notes in Artificial Intelligence, vol. 7327, Springer, Berlin, Heidelberg, s. 638–647. https://doi.org/10.1007/978-3-642-30947-2_69.

Sędziwy A., 2016, A *new approach to street lighting design*, LEUKOS, vol. 12(3), s. 151–162. https://doi.org/10.1080/15502724.2015.1080122.

Sędziwy A., Basiura A., 2018, *Energy reduction in roadway lighting achieved with novel design approach and LEDs*, LEUKOS, vol. 14(1), s. 45–51. https://doi.org/10.1080/15502724.2017.1330155.

Sędziwy A., Kotulski L., 2016a, *Towards highly energy-efficient roadway lighting*, Energies, vol. 9(4), s. 263-1–263-16. https://doi.org/10.3390/en9040263.

Sędziwy A., Kotulski L., 2016b, *Multi-agent system supporting automated large-scale photometric computations*, Entropy, vol. 18(3), 76. https://doi.org/10.3390/e18030076.

Wojnicki I., Kotulski L., 2018a, *Empirical study of how traffic intensity detector parameters influence dynamic street lighting energy consumption: A case study in Krakow, Poland*, Sustainability, vol. 10(4), 1221. https://doi.org/10.3390/su10041221.

Wojnicki I., Kotulski L., 2018b, *Improving control efficiency of dynamic street lighting by utilizing the dual graph grammar concept*, Energies, vol. 11(2), 402. https://doi.org/10.3390/en11020402.

Wojnicki I., Ernst S., Kotulski L., Sędziwy A., 2014, *Advanced street lighting control*, Expert Systems with Applications, vol. 41, s. 999–1005. https://doi.org/10.1016/j.eswa.2013.07.044.

Wojnicki I., Ernst S., Kotulski L., 2016a, *Economic impact of intelligent dynamic control in urban outdoor lighting*, Energies, vol. 9(5), s. 314-1–314-14. https://doi.org/10.3390/en9050314.

Wojnicki I., Kotulski L., Sędziwy A., Ernst S., Strug B., 2016b, *Two-level agent environment for intelligent lighting control*, International Journal of Materials and Product Technology, vol. 53, s. 187–201. https://doi.org/10.1504/IJMPT.2016.076428.

Wojnicki I., Kotulski L., Sędziwy A., Ernst S., 2017, *Application of distributed graph transformations to automated generation of control patterns for intelligent lighting systems*, Journal of Computational Science, vol. 23, s. 20–30. https://doi.org/10.1016/j.jocs.2017.09.011.

Wojnicki I., Komnata K., Kotulski L., 2019, *Comparative study of road lighting efficiency in the context of CEN/TR 13201 2004 and 2014 lighting standards and dynamic control*, Energies, vol. 12(8), 1524. https://doi.org/10.3390/en12081524.