

Adam Sędziwy*

Transformacja danych GIS do reprezentacji grafowej dla przetwarzania agentowego

1. Wprowadzenie

Dostępność zdigitalizowanych danych kartograficznych zarówno w skali makro (miasto), jak i mikro (pojedyncze budynki), umożliwia wielorakie ich wykorzystanie. Zastosowanie znajdują one w takich dziedzinach jak administracja czy zarządzanie nieruchomościami (gruntami). Również przy ich wykorzystaniu działają usługi sieciowe typu *web mapping* (Openstreetmap, Google Maps). Jeszcze innym potencjalnym obszarem zastosowania dla cyfrowych danych GIS jest projektowanie różnorodnych rozwiązań działających w przestrzeni miejskiej, takich jak: systemy ewakuacyjne, systemy kontroli natężenia hałasu czy oświetlenia. Ostatni z wymienionych przykładów stanowi treść niniejszego artykułu.

Celem projektowania oświetlenia jest znalezienie rozkładu punktów świetlnych, takich jak lampy, panele LED itp., w danym obszarze (mieście), zapewniającego odpowiedni poziom oświetlenia i spełniającego ewentualne dodatkowe kryteria np. dotyczące kosztów eksploatacji. Należy zauważyć, że kryterium to ma charakter lokalny, tzn. w ogólnym przypadku może być związane z niewielkim obszarem o skali rzędu od kilku do kilkudziesięciu metrów. Zagadnienie to jest w najprostszej swojej odmianie problemem statycznym, tj. raz ustalony rozkład punktów świetlnych nie ulega zmianom. W wersji dynamicznej zakłada się, iż modyfikację otrzymanego rozkładu wymuszać mogą takie czynniki jak zmiennej poziom natężenia światła dziennego, dynamiczne natężenie ruchu ulicznego czy inne. W takiej sytuacji system musi być zdolny do adaptacji do nowych warunków.

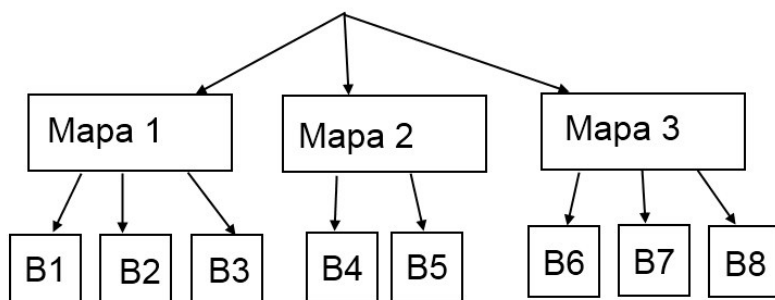
Tak postawiony problem optymalizacyjny cechuje się wysoką złożonością obliczeniową. Aby umożliwić jego rozwiązanie, należy nadać mu reprezentację, która z jednej strony stanowiłaby naturalny opis omawianego zagadnienia, z drugiej zaś, umożliwiałaby łatwą realizację zadań obliczeniowych, również w środowisku rozproszonym. Taką reprezentację stanowi model grafowy. Poza rozproszeniem samego modelu, pozwala on na stosowanie rozproszonych transformacji grafowych, opisujących dynamikę systemu [6, 7].

* AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki, al. A. Mickiewicza 30, 30-059 Kraków

Szczególnym obszarem zastosowania rozproszonej reprezentacji grafowej jest system wieloagentowy. Bazuje on na hipergrafach, których generacja na podstawie danych kartograficznych jest celem niniejszej pracy. Odpowiednia dekompozycja struktury grafowej [1, 4] umożliwi optymalną alokację zadań obliczeniowych w tym środowisku.

Należy zauważyć, że w grafowym modelowaniu danych kartograficznych dominują zwykłe grafy (skierowane lub nie). Reprezentacja hipergrafowa wprowadzona w niniejszej pracy stanowi nowe podejście. Wybór takiego właśnie opisu wiąże się z jego planowanym zastosowaniem do problemów projektowania.

Struktura problemu narzuca jego hierarchiczny opis. I tak, na wyższym poziomie umieścić można przestrzeń miejską, reprezentowaną jednolitą mapą (bądź atlasem map), będącą zbiorem ulic, placów itp. Na niższym poziomie opisu znajdują się budynki i inne obiekty, które przypisać można, z uwagi na lokalizację, określonym ulicom, a więc obiektom wyższego rzędu (rys. 1).



Rys. 1. Hierarchiczna struktura opisu

Celem niniejszej pracy jest stworzenie efektywnego, w sensie złożoności obliczeniowej, algorytmu (pseudokodu) transformacji danych OSM do reprezentacji hipergrafowej. Za rozwiązanie efektywne przyjmuje się rozwiązanie o złożoności wielomianowej.

Struktura artykułu jest następująca. W rozdziale 2 podano niezbędną, dla dalszych rozważań, charakterystykę używanych map, a ściślej, formatu w jakich są one zapisywane. W rozdziale 3 zdefiniowano pojęcie hipergrafu kartograficznego będącego docelową reprezentacją danych kartograficznych, przetwarzanych w systemie agentowym. Rozdział czwarty zawiera algorytm generacji hipergrafów kartograficznych na bazie map OSM. W ostatnim rozdziale zamieszczono podsumowanie pracy.

2. Mapy

Mapy, które będą poddawane przekształceniu do reprezentacji grafowej pochodzą z serwisu OpenStreetMap¹ [9]. Format OSM wykorzystywany do zapisu danych przez

¹ Potencjalnymi źródłami danych (w postaci map, zdjęć satelitarnych, punktów GPS itd.) dla tego serwisu są zasoby udostępniane przez instytucje, organizacje, a także osoby prywatne.

OpenStreetMap bazuje na języku XML. Elementem głównym każdej mapy jest `<osm>`. Lista dopuszczalnych typów elementów potomnych, opisujących fizyczne cechy terenu, infrastrukturę, a także niosących dodatkowe informacje, jest otwarta, choć dla zachowania zgodności danych i interoperacyjności środowiska istnieje specyfikacja cech map, opisanych standardowym zestawem tagów [10].

Mapy w standardzie OSM zbudowane są z trzech typów podstawowych, tzw. prymitywów danych: węzłów, dróg i relacji. Wszelkie obiekty, takie jak linie kolejowe, światła sygnalizacji drogowej, przystanki autobusowe itd., są złożone z prymitywów danych i tagów określających charakter obiektu, bazujących na specyfikacji cech map. Poniżej przedstawiona zostanie krótka charakterystyka prymitywów wraz z przykładami.

- 1) **Węzeł** (*node*) jest elementarnym obiektem zawierającym informację numeryczną dotyczącą długości i szerokości geograficznej swojego położenia. Może zawierać także tagi opisujące dodatkowe cechy wskazywanego miejsca. Tabela 1 prezentuje charakterystykę węzła.

Tabela 1
Struktura elementu *node*

Nazwa	Opis
id	Identyfikator węzła, unikalny wśród innych węzłów
lat	Szerokość geograficzna
lon	Długość geograficzna
tagi	Zbiór tagów postaci klucz/wartość (z niepowtarzającym się kluczem) dla opisanego punktu

Przykład ilustrujący oznaczenie świateł sygnalizacji ulicznej za pomocą prymitywu *node*:

```
<node id="25471183" lat="51.5173639" lon="-0.040043">
  <tag k="highway" v="traffic_signals"/>
</node>
```

- 2) **Droga** (*way*) jest listą co najmniej dwóch i nie więcej niż 2000 węzłów. Za jej pomocą opisać można takie obiekty, jak ulice, ścieżki, linie przesyłowe, rzeki, ogrodzenia itp. Szczególnym przypadkiem drogi jest droga zamknięta (pierwszy i ostatni węzeł są jednakowe), która pozwala na opisanie określonego obszaru, np. jeziora (zob. przykład poniżej). Tabela 2 specyfikuje strukturę elementu drogi.

Tabela 2
Struktura elementu *way*

Nazwa	Opis
id	Identyfikator drogi
węzły	Wszystkie węzły tworzące drogę
tagi	Zbiór tagów postaci klucz/wartość bez powtarzających się kluczy

Przykład ulicy jednokierunkowej zapisanej za pomocą prymitywu *way*:

```
<way id="5090250">
  <nd ref="822403"/>
  <nd ref="21533912"/>
  <nd ref="823771"/>
  <tag k="highway" v="residential"/>
  <tag k="name" v="Słoneczna"/>
  <tag k="oneway" v="yes"/>
</way>
```

Przykład opisu obszaru zbiornika wodnego za pomocą drogi zamkniętej:

```
<way id="4876027">
  <nd ref="31492372"/>
  <nd ref="31492338"/>
  <nd ref="31492370"/>
  <nd ref="31492371"/>
  <nd ref="31492372"/>
  <tag k="natural" v="water"/>
  <tag k="name" v="Jezioro Małe"/>
</way>
```

- 3) **Relacja** (*relation*) jest zbiorem (również pustym) prymitywów z określonymi rolami. Relacja może służyć zarówno określeniu związków między obiektami, jak i budowie obiektów abstrakcyjnych, np. definiować strefę zakazu wjazdu. Tabele 3 i 4 specyfikują strukturę elementu relacji.

Tabela 3
Struktura elementu *relation*

Nazwa	Opis
id	Identyfikator relacji
tagi	Jak w Tabeli 2
Członkowie (<i>members</i>)	Uporządkowana lista prymitywów należących do relacji wraz z przypisanymi do nich rolami

Tabela 4
Struktura elementu *member*

Nazwa	Opis
type	Typ prymitywu: node lub way
ref	Identyfikator prymitywu
role	Rola prymitywu w relacji. Częstymi rolami są multipolygon oraz route

Przykład relacji multipolygon.

```
<relation id="12">
  <member type="way" ref="2878061" role="outer"/>
  <member type="way" ref="8125153" role="inner"/>
  <member type="way" ref="8125154" role="inner"/>
  <member type="way" ref="3811966" role="" />
  <tag k="type" v="multipolygon"/>
</relation>
```

3. Reprezentacja danych

Pomimo iż naturalną reprezentacją grafową dla wielu problemów związanych z przetwarzaniem map, takich jak wyznaczanie najkrótszych ścieżek itp. jest ta, w której drogi opisane są za pomocą krawędzi, a punkty za pomocą wierzchołków, dla reprezentacji danych kartograficznych na poziomie miasta wybrano reprezentacje hipergrafową.

W opisie takim drogom odpowiadają węzły hipergrafu, natomiast punktom skrzyżowań hiperkrawędzie. Wybór takiego właśnie modelu został podjęty z uwagi na fakt, że główny ciężar przetwarzania w problemie optymalnej alokacji punktów świetlnych, przypada na ulice, czyli na poziomie reprezentacji hipergrafowej na wierzchołki. Pozwala to zatem na skuteczne rozproszenie obliczeń, analogicznie jak ma to miejsce w problemach modelowanych za pomocą zwykłego formalizmu grafowego [2, 3].

Zakłada się, że w ramach modelu uwzględnione zostaną prymitywy map OSM odpowiadające drogom (tag *highway*). Poniższa definicja wprowadza pojęcie hipergrafu kartograficznego, będącego reprezentacją omawianych map.

Definicja 1. Hipergrafem kartograficznym nazywamy piątkę $G = (V, H, lab_V, lab_H, att_V, att_H)$, gdzie V jest niepustym, skończonym zbiorem wierzchołków, $H \subset \bigcup_{i \geq 1} P_i(V)$ jest niepustym, skończonym zbiorem hiperkrawędzi, lab_V, lab_H są funkcjami etykietującymi dla wierzchołków i hiperkrawędzi, a att_V, att_H są funkcjami atrybutującymi dla wierzchołków i hiperkrawędzi.

Hiperkrawędziom ze zbioru H odpowiadają punkty wspólne przynajmniej dwóch ulic, ulicy i placu, bądź dwóch placów (a zatem prymitywów typu *way*). Każdy z tych obiektów reprezentowany jest w hipergrafie za pomocą węzła ze zbioru V .

Przykład 1. Na rysunku 2 przedstawiono fragment mapy i wyodrębnione na niej ulice (s_1, s_2, s_3), place (P_1, P_2, P_3) oraz ich punkty wspólne ($x_1, x_2, \dots, x_5$). Obiekty te zostają przekształcone do elementów hipergrafu w sposób następujący:

$$V = \{s_1, s_2, s_3, P_1, P_2, P_3\}, H = \{\{P_1, s_1\}, \{P_1, P_2\}, \{P_2, s_3\}, \{P_3, P_2\}, \{P_3, s_2\}\}$$

przy czym dana hiperkrawędź określona jest przez obiekty stykające się w punkcie x_i .



Rys. 2. Przykładowa mapa serwisu OpenStreetMap (po lewej). Wyodrębnione z niej ulice i place (po prawej) stają się węzłami hipergrafu kartograficznego, punkty skrzyżowań (czarne kropki) zostają reprezentowane jako hiperkrawędzie

Wartość atrybutu $att_v(v)$ dla węzła v zawiera skojarzone z nim informacje, odczytane z danych kartograficznych oraz wszelkie dodatkowe dane, wynikające z rozważanego modelu. Także wartość atrybutu $att_H(e)$ dla hiperkrawędzi e , oprócz danych punktu (*node*), pobranych z cyfrowej reprezentacji mapy, może zawierać dodatkowe dane, specyficzne dla badanego modelu.

4. Algorytm generacji hipergrafów

W niniejszym rozdziale przedstawiony zostanie algorytm pozwalający na uzyskanie reprezentacji hipergrafowej na podstawie mapy OSM. Wejściem dla procedury jest mapa OSM, wyjściem stanowi hipergraf reprezentujący sieć ulic opisanych na mapie. Pierwszym etapem algorytmu jest wyznaczenie zbioru wierzchołków hipergrafu, które reprezentują szeroko rozumiane ulice (fizycznie mogą być to ulice, autostrady, ścieżki, place itp.). Następnie ustalone zostają wszystkie węzły (prymitywy typu *node*) występujące w przynajmniej jednej ulicy. Węzły te będą reprezentowane za pomocą hiperkrawędzi rozpiętych pomiędzy wierzchołkami hipergrafu odpowiadającymi odpowiednim ulicom. Zarówno w pierwszej, jak i drugiej fazie ustalane są wartości funkcji etykietujących i atrybutujących węzły i hiperkrawędzie. Wartości te, takie jak koordynaty punktów czy nazwy bądź rodzaje ulic, generowane są na podstawie informacji zawartych w tagach poszczególnych prymitywów.

Poniższy pseudokod prezentuje algorytm generacji hipegrafu kartograficznego.

```

GenerujHipergrafKartograficzny(MapaOSM m) {
    H= $\emptyset$ ;
    V= $\emptyset$ ;
    foreach (w  $\in$  obiekty typu way na mapie m) {
        if (w posiada klucz highway i w posiada klucz name) {
            V= $V \cup \{\mathbf{w}\}$ ;
            attV(w) = zbiór atrybutów wężła w;
            labV(w) = etykieta wężła w;
        }
    }

    foreach (w  $\in$  V) {
        foreach (n  $\in$  węzły ulicy w) {
            if (n jest przetworzony) continue;
            S = {u  $\in$  V: u zawiera węzeł n};

            if (|S| > 1) {
                H= $H \cup \{\mathbf{S}\}$ ;
                attH(S) = zbiór atrybutów hiperkrawędzi S;
                labH(S) = etykieta hiperkrawędzi S;
            }

            Oznacz n jako przetworzony;
        }

        return G=(V,H,labV,labH,attV,attH);
    }
}

```

W algorytmie wykorzystano bibliotekę *org.openstreetmap.josm* zaimplementowaną w języku JAVA, służącą parsowaniu plików OSM i będącą składową edytora map OSM [7]. Jej metody występują *implicite* w następujących miejscach pseudokodu:

- 1) obiekty typu way na mapie **m**
- 2) węzły ulicy **w**
- 3) **S**=**{u** $\in$ V: **u** zawiera węzeł **n**}

Dla implementacji hipergrafów (ściśle rzecz biorąc, mogą być to multihipergrafy, a więc struktury dopuszczające występowanie hiperkrawędzi wielokrotnych) wykorzystano bibliotekę JUNG [8], również stworzoną w języku JAVA.

Złożoność czasowa algorytmu generacji wynosi $O(ML)$, gdzie M jest ilością ulic występujących na mapie **m**, natomiast L jest maksymalną liczbą węzłów (tj. prymitywów typu *node*) wchodzących w skład ulicy.

lizm może być użyty dla dowolnego innego formatu, który w sposób kompletny (a zatem także analityczny) opisuje rozważany obszar. Dla map OSM przedstawiono efektywną procedurę przejścia od danych OSM do reprezentacji hipergrafowej.

Hipergrafowy opis mapy miasta stanowi pierwszy krok budowy formalizmu, który pozwala na rozproszenie obliczeń i ich realizację przy użyciu systemu wieloagentowego. Hipergrafy modelujące fizyczny obszar, stanowią jeden z komponentów wiedzy, którą posiadają agenty.

Rozproszenie obliczeń pozwala na przezwycięzenie problemu wysokiej złożoności, stanowiącego istotną barierę w omawianym zadaniu projektowania oświetlenia.

Literatura

- [1] Kotulski L., Sędziwy A., *Agent Framework For Decomposing a Graph Into the Equally Sized Subgraphs*. WORLDCOMP'08 Conference, Foundations of Computer Science, 2008, 245–250.
- [2] Kotulski L., Sędziwy A., *GRADIS – the multiagent environment supported by graph transformations*. Simulation Modelling Practice and Theory, vol. 18, Issue 10, November 2010, 1515–1525.
- [3] Kotulski L., Sędziwy A., *Parallel Graph Transformations Supported by Replicated Complementary Graphs*. International Conference on Adaptive and Natural Computing Algorithms, ICANNGA'11, Part II, LNCS 6594, 2011, 254–270.
- [4] Kotulski L., Sędziwy A., *On the Effective Distribution of Knowledge Represented by Complementary Graphs*. Proc. of the 4th KES international conference on Agent and Multi-Agent Systems: Technologies and Applications, Part I (KES-AMSTA'10), Piotr Jędrzejowicz, Ngoc Thanh Nguyen, Robert J. Howlett, and Lakhmi C. Jain (Eds.). Springer-Verlag, Berlin, Heidelberg, 2010, 381–390.
- [5] Taentzer G., *Distributed Graphs and Graph Transformation*. Applied Categorical Structures, Springer Netherlands, vol. 7, issue 4, 1999, 431–462.
- [6] Taentzer G., Koch M., Fischer I., Volle V., *Distributed Graph Transformation With Application To Visual Design Of Distributed Systems*. Handbook of Graph Grammars and Computing by Graph Transformation, vol. 3: Concurrency, Parallelism, and Distribution, ss. 269–340, World Scientific.
- [7] JOSM – <http://josm.openstreetmap.de>.
- [8] JUNG – <http://jung.sourceforge.net/>.
- [9] OpenStreetMap – http://wiki.openstreetmap.org/wiki/Main_Page.
- [10] OpenStreetMap – http://wiki.openstreetmap.org/wiki/Map_Features.