

Piotr CICHOCIŃSKI  
AGH Akademia Górniczo-Hutnicza, Katedra Geomatyki

## ANALIZY SIECIOWE W BAZACH DANYCH PRZESTRZENNYCH<sup>1</sup>

**Streszczenie.** W artykule opisano działanie dostępnych w wybranych systemach zarządzania bazami danych, przestrzennych funkcji znajdowania najlepszej trasy przejazdu pomiędzy dwoma punktami. Szczegółowo przeanalizowano wymagania względem danych niezbędnych do przeprowadzenia tego typu analiz oraz procedurę odpowiedniego przygotowania zbioru danych sieciowych. Porównano systemy pod kątem przyjazności dla użytkownika, łatwości przygotowania danych, różnorodności dostępnych funkcji oraz stopnia skomplikowania wymaganych zapytań.

**Słowa kluczowe:** system informacji geograficznej (GIS), analizy sieciowe, baza danych przestrzennych, wolne oprogramowanie, normalizacja

## NETWORK ANALYSIS IN SPATIAL DATABASES

**Summary.** The paper describes the functionality available in chosen spatial database management systems to find the best route between two points. Requirements for data necessary to carry out such analysis and the procedure of proper network dataset preparation are thoroughly analysed. Both systems are compared in terms of user friendliness, ease of data preparation, the variety of available functions and the complexity of required queries.

**Keywords:** geographic information system (GIS), network analysis, spatial database, free software, standardization

---

<sup>1</sup> Praca naukowa zrealizowana w ramach badań statutowych prowadzonych w Katedrze Geomatyki Wydziału Geodezji Górniczej i Inżynierii Środowiska Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie.

## 1. Wprowadzenie

Pod pojęciem sieci należy rozumieć zbiór wzajemnie powiązanych obiektów: linii (zwanych krawędziami) – reprezentujących osie dróg i ulic, oraz punktów (węzłów) – w mniejszej skali reprezentujących miasta, w większej odpowiadających skrzyżowaniom [1]. Wzdłuż krawędzi odbywa się analizowany ruch. Węzły występują na styku dwóch lub więcej krawędzi i pozwalają na przepływ pomiędzy różnymi krawędziami. Aby sieć mogła funkcjonować jako model jak najbardziej zbliżony do świata rzeczywistego, z krawędziami musi być powiązana dodatkowa informacja, określająca, czy odcinek jest przejezdny w dwóch czy też w jednym kierunku, albo nieprzejezdny, oraz pewna miara określająca koszt (albo opór) ruchu wzdłuż odcinków sieci.

Na podstawie o tego typu danych działają funkcje analiz sieciowych w oprogramowaniu systemów informacji geograficznej (GIS), badające możliwości poruszania się wzdłuż poszczególnych elementów składowych sieci. Najszerzej wykorzystywana i najczęściej spotykana jest funkcja znajdowania optymalnej trasy pomiędzy dwoma punktami. Aby można było przeprowadzić taką analizę w systemie komputerowym, konieczne jest zbudowanie odpowiedniej bazy danych. Ze względu na specyfikę rozwiązywanego problemu baza taka, oprócz atrybutów opisowych obiektów, powinna zawierać także zapis ich geometrii – co czyni z niej bazę danych przestrzennych.

Podwaliny dla baz danych przestrzennych dały działania prowadzone na polu normalizacji w dziedzinie informacji geograficznej przez Międzynarodową Organizację Normalizacyjną (ang. *International Organization for Standardization* – ISO), których efektem jest seria norm ISO 19100. W szczególności w normie ISO 19125-2 [9] określono metody przechowywania, wyszukiwania, zapytań i aktualizacji kolekcji prostych obiektów geograficznych z poziomu języka SQL w relacyjnych bazach danych. Opracowanie formalnych podstaw dla baz danych przestrzennych spowodowało powstanie rozszerzeń tradycyjnych relacyjnych baz danych, pozwalających na zapis informacji geograficznej. W grupie wolnego oprogramowania [8], którego jedną z zalet jest możliwość jego bezpłatnego wykorzystania do dowolnych celów, bez konieczności ponoszenia kosztów zakupu licencji, można wymienić PostGIS, jako rozszerzenie baz PostgreSQL i Spatialite, będący rozszerzeniem SQLite. Oba te narzędzia dysponują również funkcjami analiz sieciowych. Tym samym bazy danych przestrzennych są w stanie przejąć część funkcjonalności oprogramowania GIS. Może to znaleźć zastosowanie szczególnie w przypadku zastosowania baz danych przestrzennych, jako jednego z podstawowych elementów konstrukcyjnych dla coraz szybciej rozwijających się infrastruktur informacji przestrzennej. Zakłada się, że do korzystania z udostępnianych w ten sposób usług wy-

starczy w wielu wypadkach zwykła przeglądarka internetowa. Jednym z istotnych zadań jest znajdowanie optymalnej trasy przejazdu.

Przeprowadzenie analiz sieciowych wymaga zgromadzenia i odpowiedniego przygotowania danych. W związku dużą popularnością odbiorników GPS i szerokim ich zastosowaniem w nawigacji samochodowej, istnieje wiele firm oferujących zbiory danych sieciowych. Jednak nabycie takich danych wiąże się ze sporymi kosztami. Dlatego ciekawą alternatywą może być tutaj OpenStreetMap [7] – projekt społecznościowy mający na celu utworzenie edytowalnej i dostępnej bez ograniczeń mapy świata. Dane pochodzące z takiego źródła wymagają jednak weryfikacji, poprawienia wychwyconych błędów i uzupełnienia braków. Znaczną część z tych operacji trzeba wykonać w odpowiednim oprogramowaniu GIS, lecz przynajmniej kilka wygodnie można zrealizować również z poziomu systemu zarządzania bazą danych przestrzennych. Dysponując już właściwie przygotowanym zbiorem danych sieciowych, można sformułować zapytania w języku SQL, dające w odpowiedzi informacje na temat optymalnych tras przejazdu.

## 2. Podstawy teoretyczne i formalne

Komitet techniczny ISO/TC 211 „Geographic information/Geomatics” od 1994 roku zmierza do ustanowienia ustrukturyzowanego zbioru norm dla informacji dotyczących obiektów lub zjawisk związanych bezpośrednio bądź pośrednio z położeniem względem Ziemi. Normy te mają określać metody, narzędzia i usługi do zarządzania danymi (włączając w to definiowanie i opisywanie), pozyskiwanie, przetwarzanie, analizowanie, udostępnianie, prezentowanie i przesyłanie tych danych w postaci cyfrowej pomiędzy różnymi użytkownikami, systemami i lokalizacjami. Tym samym mają wspierać zrozumienie i wykorzystanie informacji geograficznej, zwiększać jej dostępność, integrację, wspólne wykorzystanie, pozwalając na współdziałanie systemów komputerowych wzbogaconych o możliwości geoprzestrzenne i ułatwiając tworzenie infrastruktur informacji przestrzennej na poziomach lokalnym, regionalnym i globalnym.

Funkcjonalnością, która jak na razie nie została znormalizowana, są analizy sieciowe. Mimo tego powstają jednak narzędzia umożliwiające przeprowadzanie tego typu analiz, nie tylko w ramach pakietów oprogramowania GIS, ale również w systemach zarządzania bazami danych przestrzennych. Najprostszą i najczęściej stosowaną metodą rozwiązania problemu poszukiwania najkrótszej drogi jest algorytm opublikowany w 1959 roku przez holenderskiego naukowca Edsgera Dijkstrę [3].

W celu znalezienia najkrótszej drogi łączącej węzeł początkowy  $p$  z węzłem końcowym  $k$  algorytm Dijkstry zapisuje, w postaci jednowymiarowej tablicy, najkrótsze odległości od

$p$  do kolejno analizowanych węzłów. Z tablicy wybierany jest węzeł z przypisaną najmniejszą odległością od  $p$  i obliczane są odległości do wszystkich jego sąsiadów, dla których ta odległość wcześniej nie została wyznaczona. Proces wyboru węzła (spośród sąsiadów) i obliczania kolejnych odległości jest powtarzany do momentu dotarcia do węzła  $k$ .

Algorytm Dijkstry jest szczególnym przypadkiem bardziej ogólnego algorytmu  $A^*$  ( $A$  z gwiazdką – ang. *A star*), którego celem jest optymalizacja kolejności przeszukiwanych węzłów. W odróżnieniu od algorytmu Dijkstry, przeglądającego węzły w kolejności losowej, algorytm  $A^*$  oblicza odległość euklidesową wybieranego węzła od docelowego  $k$  i dodaje ją do bieżącej, wyznaczonej najkrótszej odległości. Węzły charakteryzujące się najmniejszą sumaryczną odległością są odwiedzane w pierwszej kolejności. Tym samym algorytm  $A^*$  analizuje mniejszą liczbę węzłów i z reguły jest znacznie szybszy od innych rozwiązań [2].

### 3. Wybrane implementacje baz danych przestrzennych

PostgreSQL to jeden z najpopularniejszych wolnodostępnych systemów zarządzania relacyjnymi bazami danych. Jego cechą charakterystyczną jest możliwość definiowania przez użytkowników własnych typów danych. Cecha ta została wykorzystana do zdefiniowania typów dla danych przestrzennych, będących podstawą modułu PostGIS. W efekcie PostGIS rozszerza bazę PostgreSQL o czynnik przestrzenny, pozwalając tym samym na jej wykorzystanie jako źródła danych dla oprogramowania systemów informacji geograficznej (GIS). PostGIS został opracowany przez firmę Refrations Research i jest udostępniany na licencji GNU GPL. Oprogramowanie to jest wciąż intensywnie rozwijane i jego możliwości już wykraczają poza wymagania określone w normach ISO.

Jednym z takich niestandardowych rozszerzeń jest moduł pgRouting, wzbogacający PostGIS o możliwość wykonywania analiz sieciowych. Instalacja pgRouting w systemie operacyjnym Windows polega na wgraniu do folderów oprogramowania PostgreSQL/PostGIS odpowiednich bibliotek DLL oraz uruchomieniu w bazie, w której mają być przeprowadzane analizy, trzech skryptów SQL, definiujących właściwe funkcje.

Spatialite jest rozwijany przez Alessandra Furieriego, na podstawie innego, również jednoosobowego, projektu – SQLite. SQLite jest tak naprawdę biblioteką implementującą samowystarczalny, niewymagający środowiska serwerowego i zaawansowanej konfiguracji, silnik transakcyjnej bazy danych, zarządzanej przez polecenia języka SQL. Z powodu operowania na pojedynczych plikach, w pewnym stopniu można go porównać do Microsoft Access. Kod źródłowy SQLite jest udostępniany na zasadzie *public domain*. SQLite jest najbardziej rozpowszechnioną na świecie bazą danych. Można ją znaleźć w takich popularnych

programach, jak Mozilla Firefox czy Adobe Photoshop. Niepotwierdzone informacje mówią także o wykorzystaniu SQLite w produktach firmy Apple: iPhone i iPod.

Spatialite rozszerza SQLite o możliwość przechowywania geometrycznych cech obiektów oraz wykonywania na podstawie tych obiektów, zapytań z użyciem funkcji przestrzennych. W praktyce składa się z kilku programów realizujących różne specyficzne zadania, uruchamiane z wiersza poleceń. Ponadto, do dyspozycji są dwa narzędzia o charakterze bardziej ogólnym, dysponujące graficznym interfejsem użytkownika: `spatialite_gui` i `spatialite_gis`. Pierwszy z nich jest typowym programem do zarządzania bazą danych, pozwalającym na tworzenie i edycję baz danych oraz formułowanie zapytań w języku SQL i wyświetlanie uzyskanych odpowiedzi. Natomiast program `spatialite_gis` jest prostą przeglądarką danych przestrzennych, umożliwiającą również wizualizowanie wyników zapytań w postaci nieskomplikowanych map. Zarówno w jednym, jak i w drugim programie istnieje możliwość skorzystania z funkcji analiz sieciowych, będących jednym z podstawowych komponentów Spatialite.

#### 4. Przygotowanie zbioru danych sieciowych

Jako dane do analizy wykorzystano posiadany w Katedrze Geomatyki zbiór ulic ścisłego centrum Starego Miasta w Krakowie, otoczonego Plantami. Był on wcześniej wykorzystywany w innych projektach, a w przypadku analiz sieciowych jest szczególnie dobrym przykładem, ze względu na dużą liczbę ulic jednokierunkowych i nieprzejezdnych.

Zbiór linii, wystarczający do utworzenia prezentacji w postaci mapy, najczęściej nie tworzy sieci w ścisłym tego słowa znaczeniu. W celu przeprowadzenia różnych analiz sieciowych niezbędny jest jeszcze zbiór węzłów – punktów zlokalizowanych w miejscach, gdzie stykają się ze sobą poszczególne linie. W praktyce węzły te nie muszą fizycznie istnieć, ważne jest tylko określenie, w postaci atrybutów (czyli pól w tabeli) numerów węzłów początkowych i końcowych dla każdej linii, spójnych dla wszystkich linii spotykających się w danym punkcie. Istnieją odpowiednie narzędzia w programach GIS, umożliwiające określenie numeracji węzłów i zapisanie ich we właściwych polach tabeli.

Zadanie takie można jednak w dosyć prosty sposób wykonać już/również w bazie danych z poziomu języka SQL. W dużej części, niezależną od konkretnego systemu zarządzania bazą danych, propozycję sposobu rozwiązania tego problemu można znaleźć na stronie [6]. Natomiast moduł `pgRouting` dysponuje specjalną funkcją `assign_vertex_id()`, realizującą to zadanie.

Innym ograniczeniem w wykorzystaniu różnych zbiorów dla celów analiz sieciowych są wzajemne relacje przestrzenne pomiędzy liniami – kiedy koniec jednej linii nie styka się precyzyjnie z końcami innych linii w danym zbiorze. Ponadto, jeżeli linie przecinają się ze sobą,

należy określić, czy jest to zamierzone, czy też powinno być potraktowane jako błąd. Często oprogramowanie GIS „naprawia” wszelkie przecięcia linii, dokonując ich podziału na mniejsze części, stykające się ze sobą końcami. Powstaje w ten sposób graf płaski. Jednak są sytuacje, gdy jest to niepożądane, na przykład przy modelowaniu skrzyżowań wielopoziomowych [5]. Dlatego każdy taki przypadek powinien być indywidualnie rozpatrzony. Przypadki drobnych niedokładności w powiązaniu ze sobą linii, wynikające na przykład z zaokrągleń współrzędnych, mogą być automatycznie poprawione przy założeniu, że jeżeli dwa końce linii są od siebie odległe nie więcej niż o pewną wartość tolerancji, to powinny być uznane za tożsame i z pewnością tworzą jeden węzeł.

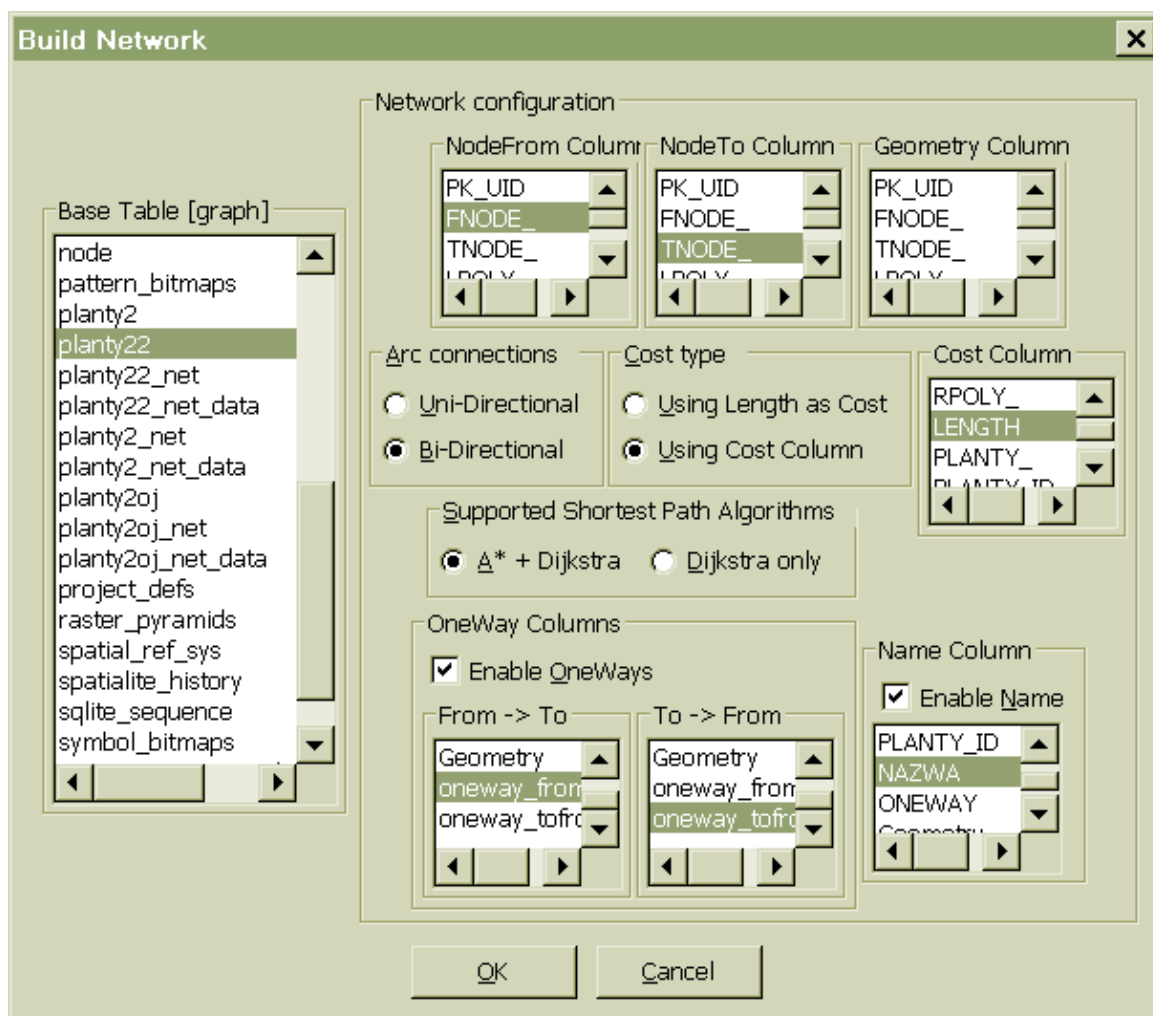
Warto, żeby jednym z atrybutów odcinków ulic była ich nazwa. Ułatwia to późniejsze przygotowanie opisu znalezionej najkrótszej trasy, szczególnie w przypadkach, gdy jej prezentacja w postaci mapy byłaby niewystarczająca.

Aby umożliwić przeprowadzenie takich analiz, konieczne jest przypisanie poszczególnym odcinkom linii atrybutów opisujących koszt przemieszczenia się wzdłuż danego elementu, w zależności od kontekstu, zwany oporem (ang. *impedance*) lub zapotrzebowaniem (ang. *demand*). Podstawową i najłatwiejszą do uzyskania miarą jest długość odcinka, która może być w prosty sposób wyznaczona na podstawie geometrii obiektu. Inną istotną cechą jest czas niezbędny do pokonania danego fragmentu sieci, będący ilorazem długości oraz prędkości przemieszczania się. Wartość tego atrybutu jest już trudniejsza do uzyskania, gdyż konieczna jest znajomość prędkości. Dodatkowo, dla większego urealnienia analizy można przypisać węzłom informacje o kosztach skrętów (ang. *turn*) pomiędzy poszczególnymi odcinkami ulic spotykającymi się na danym skrzyżowaniu.

W przypadku modułu pgRouting zbiór danych sieciowych dla algorytmu Dijkstry składa się jedynie z linii, których atrybutami są numery węzłów początkowych i końcowych tych linii. W przypadku algorytmu A\* dodatkowo w czterech kolumnach muszą być zapisane współrzędne  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$  tych węzłów. Jedną z zalet takiego rozwiązania jest błyskawiczne odzwierciedlenie zmian, które zaszły w zawartości bazy danych i możliwość natychmiastowego uwzględnienia ich w przeprowadzanych analizach, pod warunkiem oczywiście, że nie zostały naruszone wzajemne powiązania przestrzenne pomiędzy poszczególnymi odcinkami ulic.

Natomiast w systemie SpatiaLite trzeba najpierw wygenerować „wirtualną sieć” (*VirtualNetwork*) zapisaną w postaci tabeli z przyrostkiem `_net` w nazwie. Służące do tego celu odpowiednie narzędzie (rys. 1) pozwala określić, na podstawie jakiego zbioru danych wirtualna sieć będzie budowana, w jakich kolumnach zapisane są numery węzłów, jaka kolumna opisuje koszt ruchu, gdzie jest zapisana nazwa ulicy oraz, gdy uwzględniane są ulice jednokierunkowe, jaka kolumna informuje o przejeździe w kierunku zgodnym (od węzła początkowego dla danej linii do końcowego), a jaka w przeciwnym (od węzła końcowego do po-

czątkowego). Można również w tym miejscu wybrać preferowany algorytm: Dijkstry lub A\*. Wadą takiego podejścia jest to, że wirtualna sieć musi być za każdym razem zbudowana od nowa po dokonaniu jakichkolwiek zmian w tabeli, na podstawie której powstała.



Rys. 1. Narzędzie do budowania „wirtualnej sieci” w systemie zarządzania bazą danych przestrzennych SpatiaLite

Fig. 1. Tool for “virtual network” building in SpatiaLite spatial database management system

Problem kosztu i ulic jednokierunkowych również został różnie rozwiązany w tych dwóch programach. W pgRouting mogą występować dwie kolumny. Jedna, o nazwie *cost*, zawiera koszt zgodny lub ogólny, w przypadku rozważania wyłącznie ruchu dwukierunkowego. Natomiast druga, opcjonalna, o nazwie *reverse\_cost*, może zawierać koszt przeciwny. Blokowanie przejazdu w jednym bądź obu kierunkach polega na nadaniu odpowiedniemu polu kosztu wartości bardzo dużej (o wiele większej od typowego kosztu odcinka przejezdnego). W opisywanym przykładzie zamiast długości rzędu kilkudziesięciu bądź kilkuset metrów wpisano 1 000 000 (jeden milion). Wydaje się, że również skutecznym rozwiązaniem byłoby pomnożenie kosztu wyznaczonego dla danego odcinka przez 1000.

Zastosowane w pgRouting rozwiązanie nie jest jednak do końca poprawne, gdyż z braku możliwości wyznaczenia innej trasy w ostateczności ruch może zostać zaplanowany nawet drogą scharakteryzowaną jako nieprzejezdna. Teoretycznie program dopuszcza zapis ujemnego kosztu, co ma powodować wykluczenie danego odcinka z analizy, lecz w praktyce skutkowało to jedynie zawieszaniem się programu. Być może wynikało to z zastosowania starszej wersji pgRouting, lecz tylko taka jest dostępna na komputery z systemem operacyjnym Windows. Zostanie to sprawdzone w toku dalszych badań, po przygotowaniu odpowiedniego stanowiska komputerowego z systemem operacyjnym Linux.

Natomiast w SpatiaLite wartość 1 lub 0 w dwóch kolumnach typu *integer* informuje odpowiednio o przejezdności bądź nieprzejezdności danego odcinka w kierunku zgodnym i przeciwnym. Wadą takiego rozwiązania jest to, że koszt ruchu musi być taki sam w obu kierunkach. Co nie zawsze musi być prawdą, na przykład, gdy rejestrowany jest wzmożony ruch samochodów tylko w jednym kierunku, powodujący powstawanie „korków”.

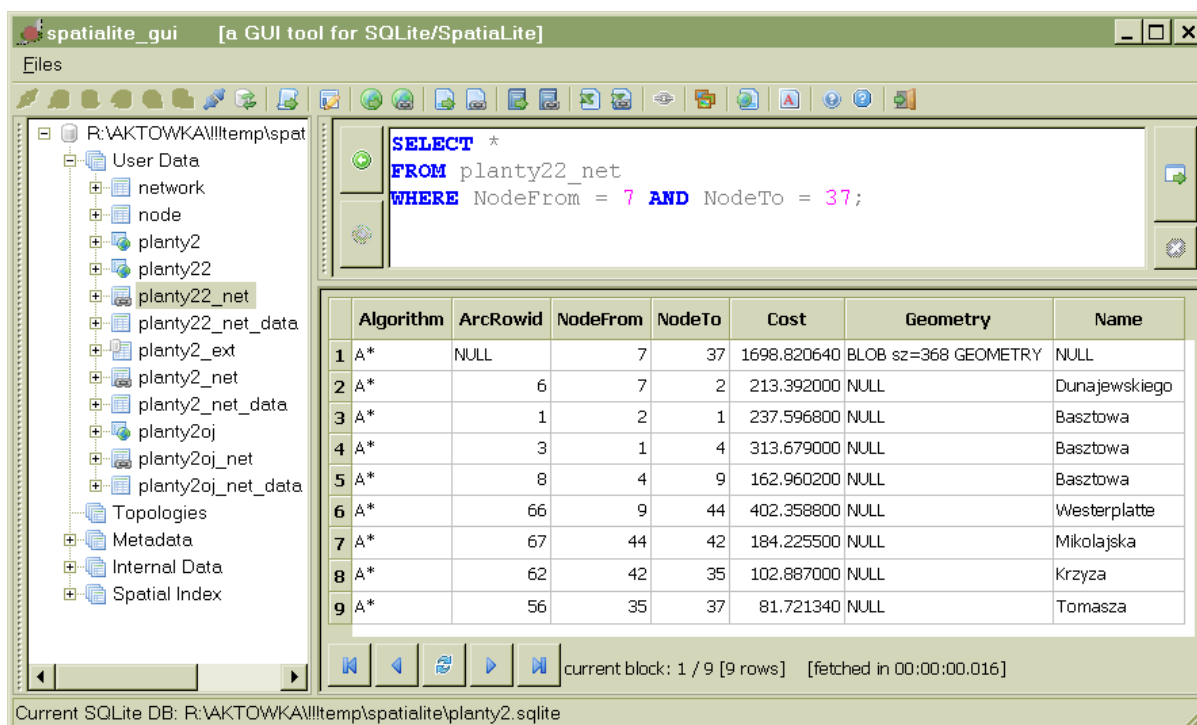
## 5. Przebieg i wyniki analiz sieciowych

Jako przykład dobrze ilustrujący i weryfikujący możliwości opisanych w pracy baz danych przestrzennych i ich rozszerzeń, umożliwiających wykonywanie analiz sieciowych, postanowiono wyznaczyć najkrótszą trasę przejazdu pomiędzy dwoma końcami ulicy św. Tomasza. Ulica ta jest w większości jednokierunkowa, co sugeruje konieczność poszukiwania objazdu przy chęci ruchu „pod prąd”. Dodatkowo sąsiednie ulice też są jednokierunkowe lub wręcz nawet nieprzejezdne, tak jak te zlokalizowane na Rynku Głównym, co zapewne spowoduje dalsze wydłużenie trasy.

Przeprowadzenie analizy sieciowej w systemie zarządzania bazą danych polega na sformułowaniu odpowiedniego zapytania zapisanego w języku SQL. Wynikiem zapytania jest tabela zawierająca listę kolejnych odcinków ulic, które trzeba przejechać w drodze od węzła początkowego  $p$  do węzła końcowego  $k$  (rys. 2). Opierając się na tym przykładzie, można sądzić, że nie jest konieczne posiadanie żadnego interfejsu graficznego, aby przeprowadzać analizy sieciowe. Pewnym utrudnieniem może być tylko konieczność podania numerów tych węzłów w zapytaniu. Typowym dla programów GIS sposobem określania punktów charakterystycznych jest wskazywanie ich na mapie za pomocą kursora. Niestety z reguły w oprogramowaniu zarządzającym bazą danych nie jest to możliwe, czyli jednak nie da się tak naprawdę wykonać analizy bez graficznej wizualizacji danych i wyników. Przekonał się o tym autor, próbując porównać wyniki uzyskane w module pgRouting z uwzględnieniem ulic jednokierunkowych i bez. Na pierwszy rzut oka wydawało się, że testowany mechanizm nie działa



poprawnie, gdyż za każdym razem w wyniku uzyskiwano tyle samo wierszy (odcinków ulic), ponadto program nie podaje w wynikowej tabeli nazw ulic. Dopiero po wykonaniu graficznej prezentacji wyników okazało się, że są to zupełnie różne trasy i pomimo że jedna z nich była znacząco dłuższa, to składała się z takiej samej jak druga liczby odcinków.

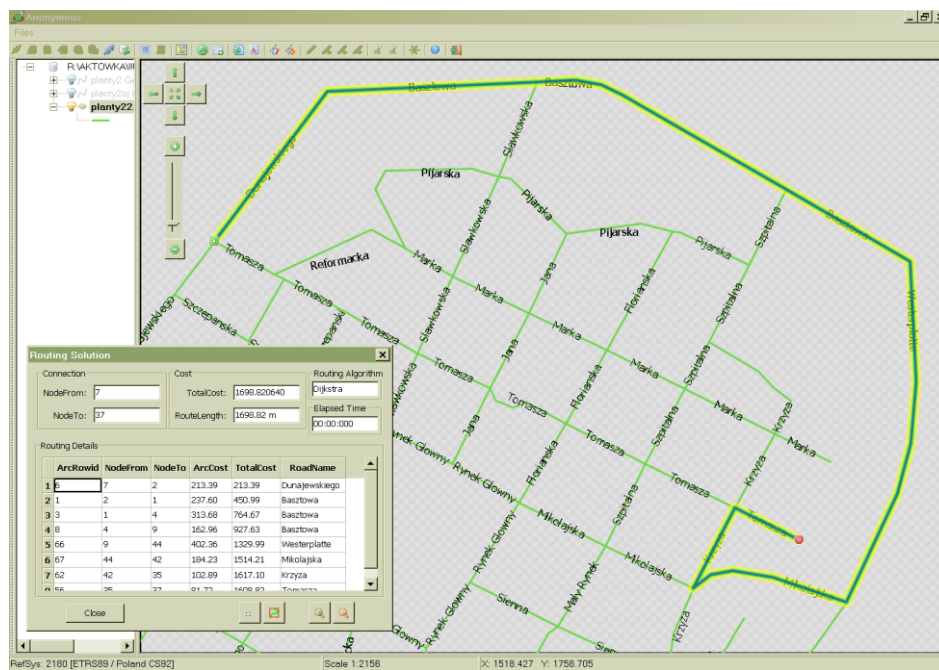


Rys. 2. Wynik zapytania SQL znajdującego najlepszą trasę pomiędzy dwoma węzłami w systemie zarządzania bazą danych przestrzennych Spatialite

Fig. 2. Result of the SQL query that finds best route between two nodes in Spatialite spatial database management system

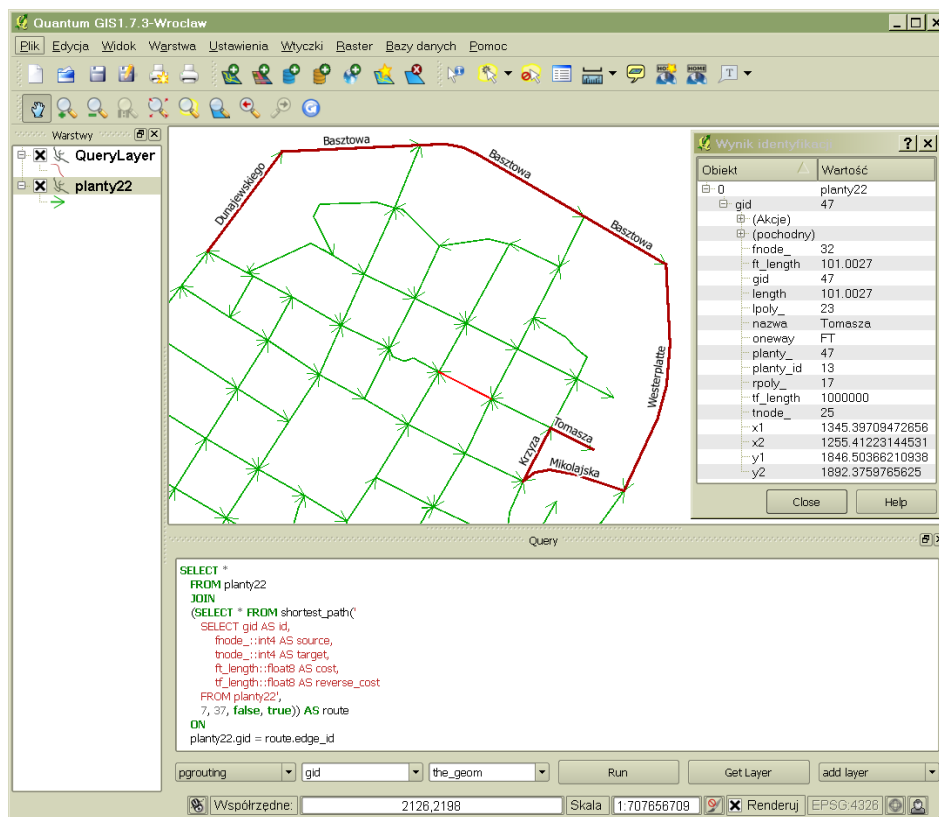
Zapewne dlatego Spatialite dysponuje prostą przeglądarką umożliwiającą łatwą prezentację mapową danych zapisanych w bazie oraz ułatwiającą wykonywanie analiz sieciowych. Pozwala bowiem w sposób graficzny określić punkty startowy i końcowy poszukiwanej najkrótszej trasy oraz natychmiast zwizualizować uzyskany wynik (rys. 3).

Takiego narzędzia nie posiada PostGIS, ale rozwiązaniem tego problemu jest dobra współpraca obu opisywanych baz danych (a w szczególności PostGIS-a) z wieloma programami GIS. Jednym z nich jest QuantumGIS, pomyślany pierwotnie jako przeglądarka danych przestrzennych i łatwy w obsłudze interfejs do współpracy z zaawansowanymi narzędziami przetwarzania danych, wchodzącymi w skład pakietu GRASS. Jednak wyposażenie QuantumGIS-a w mechanizm wtyczek (ang. *plug-in*) pozwoliło na znaczące poszerzenie jego funkcjonalności i tym samym spopularyzowanie w wielu obszarach zastosowań. Jedną z dostępnych wtyczek jest Fast SQL Layer autorstwa Pabla Carreiry. Umożliwia ona wizualizowanie wyników zapytań SQL posiadających komponent przestrzenny (rys. 4).



Rys. 3. Prezentacja najlepszej trasy w postaci tabelarycznej i postaci graficznej w programie spatialite\_gis

Fig. 3. Best route presentation in tabular and graphical form in spatialite\_gis software



Rys. 4. Wynik zapytania SQL znajdującego najlepszą trasę pomiędzy dwoma węzłami w bazie danych przestrzennych PostGIS, zwizualizowany w programie QuantumGIS

Fig. 4. Result of the SQL query that finds best route between two nodes in PostGIS spatial database, visualized in QuantumGIS software

## 6. Wnioski

Od kilkudziesięciu już lat jesteśmy świadkami ciągłego rozwoju technologii informatycznych, z którymi systemy informacji geograficznej są w dużej mierze związane – czym bowiem byłby GIS bez komputera, bazy danych i odpowiedniego systemu jej obsługi. Rozwój ten doprowadził do powstania baz danych przestrzennych, pozwalających nie tylko na zapisywanie informacji geograficznej, ale również jej przetwarzanie i analizowanie. Jednym ze specyficznych dla systemów informacji geograficznej rodzajów analiz są analizy sieciowe.

Przeprowadzone badania wybranych dwóch systemów zarządzania bazami danych przestrzennych potwierdziły użyteczność opisywanych narzędzi do wykonywania analiz sieciowych. Pozwoliły odkryć ich słabe punkty, ale pokazały również mocne strony. Wydaje się, że każdy z porównywanych systemów może znaleźć dla siebie obszary, gdzie będzie się sprawował lepiej od konkurenta.

Do podstawowych wad przeprowadzania analiz sieciowych w środowisku typowego systemu zarządzania bazą danych należy zaliczyć brak możliwości wizualizacji uzyskanego wyniku w postaci graficznej. Konieczna jest w tym celu współpraca z oprogramowaniem GIS, zalecana również w przypadkach, gdy pożądane jest wskazywanie kursorem punktów początkowych i końcowych poszukiwanych tras.

PostgreSQL/PostGIS został pomyślany jako system do obsługi dużych baz danych, działających w środowisku klient-serwer. Dysponuje odpowiednimi dla tego celu mechanizmami zabezpieczania dostępu. Zaimplementowane w tym systemie, w postaci modułu pgRouting, funkcje analiz sieciowych dodatkowo rozszerzają obszar jego zastosowań w zakresie infrastruktury informacji przestrzennej. Oprogramowanie to nie jest jeszcze wolne od błędów, takich jak niewłaściwa reakcja na ujemny koszt ruchu, ale ponieważ podlega ciągłemu rozwojowi, można mieć nadzieję, że błędy te zostaną w przyszłości wyeliminowane. Do wyróżniających pozytywnych cech modułu pgRouting można zaliczyć możliwość stosowania kosztów niezależnie dla każdego kierunku przejazdu oraz przeprowadzanie analizy bezpośrednio na odpowiednio przygotowanym zbiorze danych, co w sposób natychmiastowy powoduje uwzględnienie ewentualnych dokonanych w nim zmian. Jako wadę należy potraktować niemożność całkowitego zablokowania przejazdu, a jedynie jego ograniczenie przez określenie bardzo dużej wartości kosztu ruchu.

Spatialite jest prostym systemem bazodanowym, przeznaczonym dla pojedynczego użytkownika korzystającego z jednej aplikacji na pojedynczym komputerze. Niektórzy wieszczą nawet, że plikowy format prostej i łatwej do przenoszenia bazy danych przestrzennych zastąpi w niedalekiej przyszłości wysłużone już pliki *shape* [4]. W związku z tym nie jest przeznaczony do obsługi dużych i skomplikowanych projektów, wymagających wielozadaniowości,

ale doskonale sprawdza się przy realizacji zadań typowych dla biurkowego (ang. *desktop*) narzędzia GIS. W zakresie analiz sieciowych dysponuje dodatkowo prostą przeglądarką ułatwiającą określenie parametrów funkcji i prezentującą w postaci graficznej uzyskany wynik. Do wad tego systemu trzeba zaliczyć konieczność budowania od nowa wirtualnej sieci po dokonaniu jakichkolwiek zmian w tabeli, na podstawie której powstała, oraz brak możliwości określania różnych kosztów ruchu dla poszczególnych kierunków przejazdu.

## BIBLIOGRAFIA

1. Curtin K. M.: Network Analysis in Geographic Information Science: Review, Assessment, and Projections. *Cartography and Geographic Information Science*, Vol. 34, No. 2, 2007, s. 103÷111.
2. de Smith M. J., Goodchild M.F., Longley P. A.: *Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools* (2nd ed.). Troubador Publishing, 2007.
3. Dijkstra E. W.: A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 1959, s. 269÷271.
4. ESRI Shapefile Technical Description. An ESRI White Paper, Environmental Systems Research Institute, Redlands 1998.
5. Fischer M. M.: GIS and Network Analysis. ERSA conference papers. 43rd European Congress of the European Regional Science Association. Jyväskylä, Finland 2003.
6. Graser A.: A Beginner's Guide to pgRouting. Free and Open Source GIS Ramblings, <http://underdark.wordpress.com/2011/02/07/a-beginners-guide-to-pgrouting/>
7. Haklay M., Weber P.: OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 2008, s. 12÷18.
8. Michalak J.: Otwarte oprogramowanie i otwarte dane w geomatyce. *Roczniki Geomatyki*, Tom V, Zeszyt 2, Warszawa 2007.
9. PN-EN ISO 19125-2:2006 Informacja geograficzna – Środki dostępu do obiektów prostych – Część 2: Opcja SQL.

Wpłynęło do Redakcji 11 stycznia 2012 r.

**Abstract**

Network is a set of interconnected elements: lines (called edges) – representing road and street centerlines, and points (nodes) – in smaller scale representing cities, in greater scales corresponding to street junctions. Network analysis functions in geographic information systems (GIS) software use this type of data to explore movement possibilities along network components. Function that finds the best route between two points is the most widely used. To be able to carry out this analysis in a computer system, it is necessary to build appropriate spatial database.

Spatial databases are a kind of databases, taking into account the spatial aspect of data, which means that in addition to descriptive attributes of the objects, the database also stores their geometric properties. The formal framework for this new way of spatial data storage was laid by ISO series 19100 standards and Open Geospatial Consortium (OGC) specifications. In particular ISO 19125-2 standard specifies methods for storage, retrieval, query and update of simple feature collections, using SQL language in relational databases.

This has resulted in creation of traditional relational databases extensions, allowing geographic information storage. The following open source software can be mentioned: PostGIS as an extension to PostgreSQL, and SpatiaLite being extension to SQLite. Both tools have, among others, network analysis functions.

The paper describes the functionality available in these spatial database management systems to find the best route between two points. Requirements for data necessary to carry out such analysis and the procedure of proper network dataset preparation are thoroughly analysed. The problem of data availability for this type of analysis is mentioned and OpenStreetMap is identified as a viable alternative to commercial spatial data providers. Possibilities of controlling the course of analysis are assessed. Capabilities of query formulation and visualisation of obtained results in popular GIS software are considered. Both systems are compared in terms of user friendliness, ease of data preparation, the variety of available functions and the complexity of required queries.

**Adres**

Piotr CICHOCIŃSKI: AGH Akademia Górniczo-Hutnicza, Katedra Geomatyki,  
al. Mickiewicza 30, 30-059 Kraków, Polska, Piotr.Cichocinski@agh.edu.pl.