

Adrian OLCZYK, Adam GAŁUSZKA
Politechnika Śląska

WPLYW MODELI GRAFOWYCH NA CZAS WYSZUKIWANIA POŁĄCZEŃ W SIECI KOMUNIKACJI MIEJSKIEJ

Streszczenie.

Praca prezentuje trzy modele grafowe reprezentujące sieć komunikacji miejskiej (model stacji, model oparty na wydarzeniach i model zależny od czasu), a następnie porównuje czas wyszukiwania, czas początkowego przetwarzania i wymaganą pamięć operacyjną dla każdego z nich. Najlepsze wyniki osiągnął model stacji.

EFFECTS OF THE GRAPH MODELS ON THE SHORTEST PATH SEARCH TIME IN A PUBLIC TRANSPORTATION NETWORK

Summary. This manuscript presents three graph models representing public transportation networks (station model, time-expanded model, and time-dependent model) and compares their search time, preprocessing time, and memory requirements. The station model achieved the best results.

1. Wstęp

Algorytmy wyszukiwania drogi i połączeń w sieci komunikacji miejskiej są problemem bardzo chętnie rozpatrywanym przez naukowców [6, 3]. Natomiast niewiele prac skupia się na wpływie modeli grafowych na czas wyszukiwania połączeń. Niniejsze streszczenie przedstawia wyniki badań porównania trzech modeli grafowych reprezentujących sieć komunikacji miejskiej: *modelu opartego na wydarzeniach*, *modelu zależnego od czasu* i *modelu stacji*.

2. Modele

Modele pozwalają na reprezentację rozkładu jazdy przy pomocy grafu. Przeniesienie problemu planowania podróży do domeny teorii grafów pozwala na wykorzystanie znanych algorytmów wyszukiwania najkrótszych ścieżek. Każdy z prezentowanych w niniejszej pracy modeli pozwala na uzyskanie tych samych wyników, ale różni się ilością węzłów i krawędzi, definicją wag krawędzi, poziomem skomplikowania implementacji i dostępnością dedykowanych rozwiązań przyspieszających działanie algorytmów wyszukiwania.

Tabela 1 zawiera tabliczkę odjazdów pojazdów dwóch linii, na przykładzie której przedstawione zostaną trzy modele grafowe: *model oparty na wydarzeniach*, *model zależny od czasu* i *model stacji*.

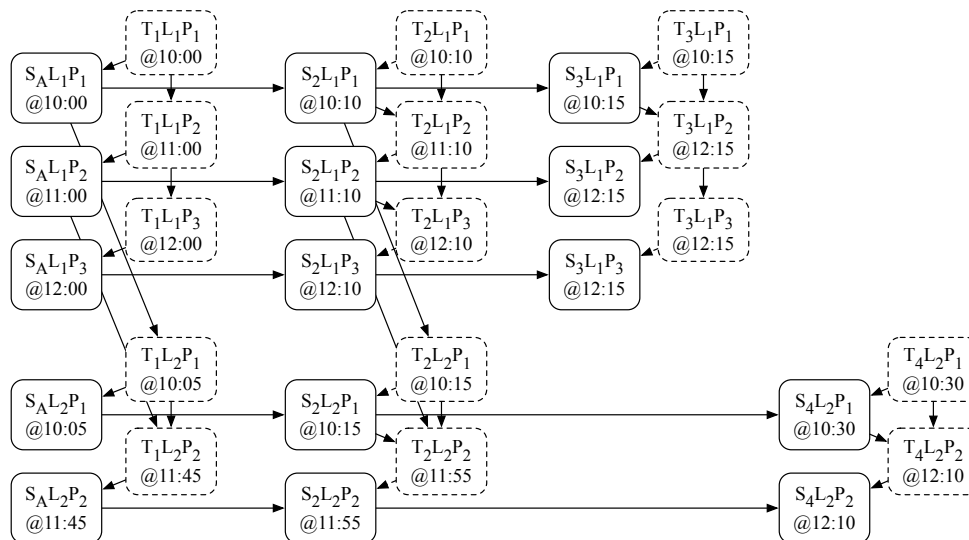
Tabela 1

Przykład tabeli odjazdów pojazdów dwóch linii. Pojazdy przemieszczają się ze stacji A do B, następnie do C i D. Puste miejsca w tabeli oznacza, że pojazd stację omija.

	Stacja A	Stacja B	Stacja C	Stacja D
Linia 1, Pojazd 1	10:00	10:10	10:15	-
Linia 1, Pojazd 2	11:00	11:10	11:15	-
Linia 1, Pojazd 3	12:00	12:10	12:15	-
Linia 2, Pojazd 1	10:05	10:15	-	10:30
Linia 2, Pojazd 2	11:45	11:55	-	12:10

2.1. Model oparty na wydarzeniach

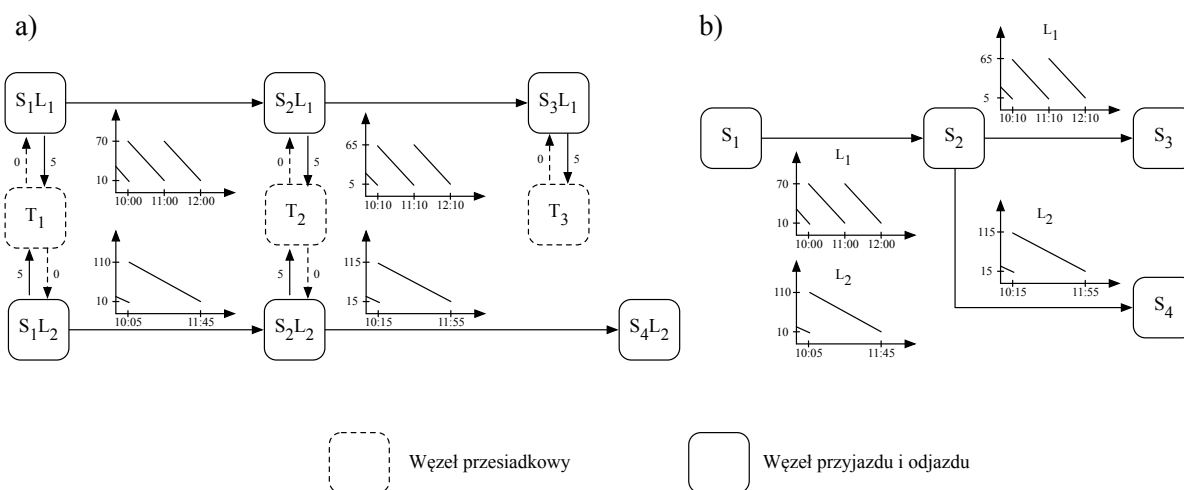
Model oparty na wydarzeniach [7, 5] zakłada, że węzeł w grafie odpowiada konkretnemu wydarzeniu elementarnemu (np. przyjazdowi lub odjazdowi pojazdu na przystanku). Krawędzie w grafie odpowiadają podróży między przystankami, a ich wagi są równe różnicy czasu między węzłami. Przesiadki realizowane są poprzez dodanie węzłów przesiadkowych (rys. 1 a)).



Rys. 1. Odjazdy pojazdów z tabeli 1 przedstawione jako: *model oparty na wydarzeniach*, gdzie L_i to linia o numerze i , S_k to stacja o numerze k , a T_k to węzeł przesiadkowy na stacji o numerze k , a P_j to pojazd o numerze j

2.2. Model zależny od czasu

Model zależny od czasu [2] ogranicza liczbę węzłów w stosunku do *modelu opartego na wydarzeniach*. W tym modelu każda unikalna trasa (np. autobusowa), wzdłuż której przemieszczają się pojazdy, posiada na przystanku tylko jeden węzeł. Wagi krawędzi są funkcją czasu, a przesiadki są realizowane poprzez dodanie węzłów przesiadkowych (rys. 2 a)).



Rys. 2. Odjazdy pojazdów z tabeli 1 przedstawione jako: a) *model zależny od czasu*, b) *model stacji*, gdzie L_i to linia o numerze i , S_k to stacja o numerze k , a T_k to węzeł przesiadkowy na stacji o numerze k , a P_j to pojazd o numerze j

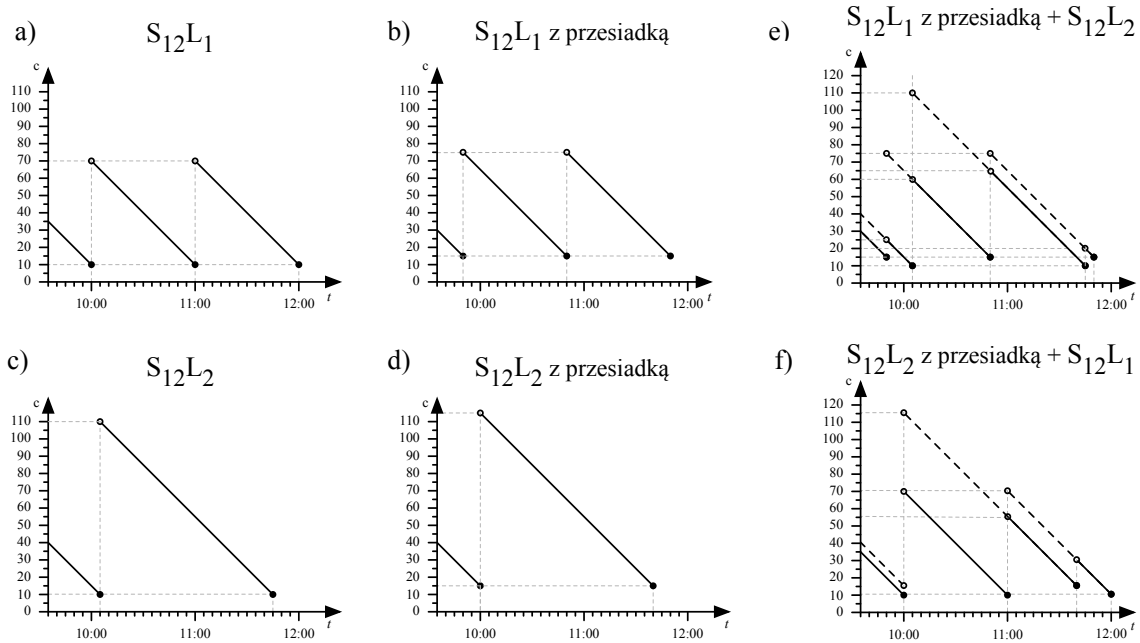
2.3. Model stacji

Model stacji jest bardzo modelem elastycznym i jednocześnie najtrudniejszym w implementacji. Zakłada, że na przystanku istnieje jedynie jeden węzeł (rys. 2 b)). Połączenia do innych węzłów tworzone są wtedy, jeśli możliwe jest dotarcie do kolejnego przystanku przy pomocy istniejącego połączenia. Wagi krawędzi są funkcją nie tylko czasu, ale i linii, którą pasażer podróżuje (rys. 3).

3. Metodologia

Praca zakłada porównanie trzech parametrów dla każdego z modeli: czasu początkowego przetwarzania, zapotrzebowania na pamięć oraz czasu wyszukiwania połączeń. Jako algorytm wyszukiwania najkrótszej ścieżki został wybrany algorytm Dijkstry [4]. Eksperymenty zostały przeprowadzone dla sieci ZTM, która jest siecią komunikacji miejskiej funkcjonującej na Śląsku. W momencie wykonywania eksperymentów sieć składała się z 2816 przystanków i 2957 linii.

Eksperymenty zostały przeprowadzone na komputerze z procesorem 3.4 GHz Quad-Core Intel Core i7 wyposażonym w 24 GB pamięci RAM. Każdy model sieci został zaimplementowany w języku JavaScript i uruchomiony po stronie serwera przy pomocy środowiska uruchomieniowego *node.js*.



Rys. 3. Wagi krawędzi dla modelu stacji łączącej przystanki S_1 i S_2 liniami L_1 i L_2 . Wykres a) i c) przedstawia wagi krawędzi dla obu linii w postaci funkcji czasu (podobnie jak w przypadku modelu zależnego od czasu). Wykres b) i d) zakłada konieczność przesiadki, która na przystanku S_1 trwa 5 minut. Wykres e) przedstawia funkcję wag krawędzi dla pasażera poruszającego się linią L_2 . Wykres f) przedstawia funkcję wag krawędzi dla pasażera poruszającego się linią L_1

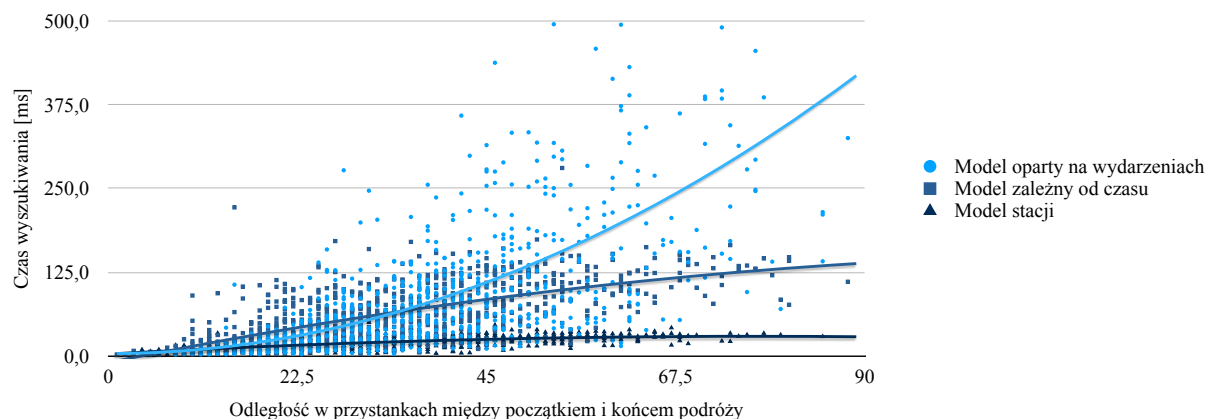
4. Wyniki

4.1. Czas wyszukiwania

Wykres punktowy (rys. 4) przedstawia zależność czasu wyszukiwania od odległości między przystankiem początkowym i końcowym dla trzech modeli omawianych w niniejszej pracy. Każdy punkt na wykresie jest osobnym pomiarem wykonanym zgodnie z metodologią. Dla każdego modelu została obliczona krzywa trendu metodą najmniejszych kwadratów. Funkcja krzywej jest wielomianem. Współczynnik determinacji R^2 dla modelu opartego na wydarzeniach, modelu zależnego od czasu i modelu stacji jest równy kolejno 0.4584, 0.4464 i 0.3524.

Czas wyszukiwania dla modelu opartego na wydarzeniach rośnie eksponencjalnie i dla sieci ZTM sięga 500 ms. Model zależny od czasu, ze względu na znacząco zmniejszoną liczbę węzłów, zachowuje się lepiej od modelu opartego na wydarzeniach. Czas wyszukiwania sięga 200 ms, ale zazwyczaj ta liczba jest znacznie mniejsza. Linia trendu bardziej przypomina zależność liniową.

Najlepsze wyniki osiągnął model stacji. Czasy wyszukiwania sięgają 100 ms, ale średnia jest znacznie niższa od pozostałych modeli. Linia trendu w pewnym momencie (przy odległości około 40 przystanków) przestaje rosnać i utrzymuje się na podobnym poziomie (średnio około 35 ms).

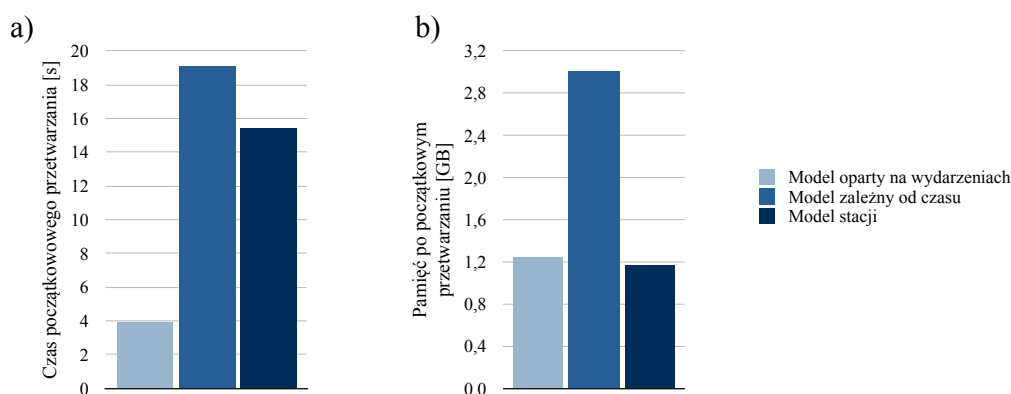


Rys. 4. Czas wyszukiwania dla *modelu opartego na wydarzeniach*, *modelu zależnego od czasu* i *modelu stacji*

4.2. Czas początkowego przetwarzania

Zwiększenie czasu potrzebnego na przygotowanie pełnego modelu sieci wpływa (negatywnie lub pozytywnie) na zapotrzebowanie pamięci (rys. 5 a)) i czas wyszukiwania najkrótszej ścieżki.

Model oparty na wydarzeniach jest najprostszym modelem, wymagającym najmniej czasu na przygotowanie. *Model oparty na czasie* i *model stacji* wymagają więcej obliczeń związanych z definicją węzłów, krawędzi oraz funkcji wag. W związku z tym czas początkowego przetwarzania obu tych modeli jest wyższy niż modelu opartego na czasie.



Rys. 5. a) czas początkowego przetwarzania i b) zapotrzebowanie na pamięć dla *modelu opartego na wydarzeniach*, *modelu zależnego od czasu* i *modelu stacji*

4.3. Zapotrzebowanie na pamięć

Zapotrzebowanie na pamięć (rys. 5) rośnie wraz ze złożonością modelu. Każdy model, zajmował mniej niż 4 GB pamięci. Oznacza to, że dla badanej sieci, jest możliwe utrzymywanie pełnego modelu w pamięci operacyjnej.

Model oparty na wydarzeniach zajmował najmniej pamięci, w porównaniu do

modelu opartego na wydarzeniach i zależnego od czasu. Wynika to z faktu, że *model zależny od czasu* i *model stacji* posiada krawędzie, których waga jest funkcją kilku zmiennych wejściowych (głównie czasu). Aktualna implementacja zakładająca dyskretność tych danych (czas liczony w inkrementach minutowych) optymalizuje czas obliczania wagi dla danych wejściowych poprzez przygotowanie statycznej mapy wartości wyników.

Model zależny od czasu, zajmował najwięcej pamięci. Wynika to z faktu, że konieczne jest przechowywanie większej ilości informacji o wagach krawędzi. *Model stacji* zajmuje mniej pamięci niż model zależny od czasu, ale jest znacznie bardziej zbliżony do modelu opartego na wydarzeniach.

5. Podsumowanie

Przeprowadzone badania wskazują, że wybór i specyfika modelu może znacząco wpłynąć na takie parametry jak czas wyszukiwania, czas początkowego przetwarzania i wymagana pamięć. Z badanych modeli najlepsze wyniki osiągnął *model stacji*, szczególnie jeśli chodzi o czas wyszukiwania. Jest to model, który ze względu na swoją specyfikę, jest modelem posiadającym mniejszą liczbę węzłów i krawędzi w porównaniu do pozostałych modeli, a waga krawędzi jest funkcją kilku parametrów. Powoduje to zwiększenie poziomu skomplikowania implementacji oraz potencjalne utrudnione wnioskowanie. Niemniej jednak jest to model, któremu warto się przyjrzeć w dalszych badaniach ze względu na potencjał zastosowania komercyjnego.

LITERATURA

1. Basu A., Raja B., Gracious R., and Vanajakshi L.: Dynamic trip planner for public transport using genetic algorithm. *Transport*, 35, 2020.
2. Chen B. Y., Lam W. H. K., Li Q., Sumalee A., Yan K.: Shortest Path Finding Problem in Stochastic Time-Dependent Road Networks With Stochastic First-In-First-Out Property. *IEEE Transactions on Intelligent Transportation Systems*, 14(4), 2013, p. 1907–1917.
3. Cionini A., D’Angelo G., D’Emidio M., Frigioni D., Giannakopoulou K., Paraskevopoulos A., Zaroliagis C.: Engineering graph-based models for dynamic timetable information systems. *Journal of Discrete Algorithms*, 2017, p. 46-47.
4. Dijkstra E.: A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 1959, 269–271.
5. Müller-Hannemann M. and Weihe K.: Pareto Shortest Paths is Often Feasible in Practice. *Algorithm Engineering*, 2141, 2001, (Chapter 15):185–197.
6. Peng Y., Mo Z., Liu S.: Passenger’s routes planning in stochastic common-lines’ multi-modal transportation network through integrating genetic algorithm and monte carlo simulation. *Archives of Transport*, 59, 2021.
7. Schulz F., Wagner D., Weihe K.: Dijkstra’s algorithm on-line: An empirical case study from public railroad transport. *Algorithm Engineering*, 1668, 1999, 110–123.