KOZIEŁ Grzegorz[1]

Algorytmy wyznaczania optymalnej trasy przejazdu

WSTĘP

Wyznaczenie trasy przejazdu to w dzisiejszych czasach bardzo popularne zadanie. Niemniej jednak trudno jest wyznaczyć trasę optymalną. Dostawcy różnego rodzaju usług oferują tego typu rozwiązania w urządzeniach do nawigacji czy mapach dostępnych w sieci Internet. Jednak są to rozwiązania o ograniczonej możliwości konfiguracji. Najczęściej optymalizacji sprowadza się do możliwości wyboru jednego z parametrów jako głównego kryterium. Wśród nich możemy spotkać takie wielkości jak odległość, czas przejazdu czy zużyte paliwo. Jednak coraz częściej istotne jest określenie optymalnej trasy przy uwzględnieniu zupełnie innych parametrów. Dodatkowo ważne jest jednoczesne uwzględnienie wszystkich wymaganych parametrów przy zapewnieniu, że będzie istniała możliwość określenia ich „ważności”. Tak zdefiniowane parametry pozwolą na elastyczne planowanie tras przejazdu i optymalizację działania transportu. Jest to szczególnie istotne w przypadku przedsiębiorstw spedycyjnych lub producentów czy hurtowni zaopatrujących swoich klientów w towary. W warunkach rzeczywistych coraz częściej spotykamy się z planowaniem dostaw towarów do klientów. Samochód z towarem ma za zadanie dostarczenie towaru do określonej liczby klientów. Istotne są tu przede wszystkim koszty transportu oraz jego czas. Ważne jest więc zaplanowanie trasy w taki sposób by zoptymalizować żądane czynniki. Konieczne jest zastosowanie algorytmu pozwalającego uwzględnić koszt trasy. Należy jednak zwrócić uwagę na to, że koszt trasy jest sumą kosztów pokonania poszczególnych odcinków. A koszt pokonania każdego z odcinków może być zupełnie inny w każdym z rozpatrywanych wariantów trasy. Może na to wpłynąć chociażby kierunek przejazdu danego odcinka, zwłaszcza jeżeli występują różnice wysokości pomiędzy jego końcowymi punktami. Dodatkowo planując łańcuch dostaw musimy uwzględnić zwiększony koszt transportu na początku trasy, wynikający z większej masy ładunku. Parametr ten jest istotny zwłaszcza przy planowaniu transportu towarów posiadających dużą masę. Konieczne jest więc używanie algorytmów, które pozwalają na określenie kosztu przebycia każdego odcinka niezależnie w dwie strony. Ponadto algorytmy te muszą pozwalać na elastyczne wprowadzanie modyfikacji swojej metody działania w celu dostosowania ich działania do specjalnych potrzeb użytkowników. Według opinii autora, możliwe jest wprowadzenie takiej modyfikacji istniejących algorytmów wyznaczania drogi, by możliwe było uwzględnienie wielu parametrów podczas jej wyboru.

1. ALGORYTMY WYZNACZANIA DROGI

Wyznaczenie drogi możliwe jest za pomocą różnego rodzaju algorytmów przeszukiwania. Zasada działania tych algorytmów opiera się na przeszukiwaniu grafu [7]. Mapa drogowa przekształcana jest w graf skierowany [1]. Jego węzły odpowiadają punktom na mapie – najczęściej skrzyżowaniom lub miejscowościom (w zależności od skali rozpatrywanego problemu i jego szczegółowości) a krawędzie reprezentują drogi łączące poszczególne punkty. Najprostsze techniki opierają się o budowanie drzewa przeszukiwań, przeszukując je wszerz, w głąb lub przy pomocy innej techniki. W rzeczywistych warunkach nie są one jednak wystarczające, ze względu na brak możliwości uwzględnienia kosztu przejścia pomiędzy węzłami drzewa. Odpowiada to sytuacji w której koszt przejazdu pomiędzy dwoma dowolnymi punktami jest jednakowy. W rzeczywistości sytuacje takie nie występują. Konieczne jest więc korzystanie z algorytmów pozwalających na określanie kosztu przebycia drogi, które na dodatek pozwalają niezależnie określić koszt przebycia drogi w dwie strony

[1] Instytut Informatyki, Wydział Elektrotechniki i Informatyki, Politechnika Lubelska, Nadbystrzycka 36b, 20-618 Lublin
e-mail: g.koziel@pollub.pl, tel. (81) 538-46-08

oraz określić czy możliwy jest przejazd daną drogą w wybranym kierunku. W innym przypadku nie istniała by możliwość prawidłowego zinterpretowania faktu istnienia dróg jednokierunkowych oraz kosztów przebycia drogi zależnych od kierunku przejazdu, takich jak chociażby różnica poziomów pomiędzy punktami krańcowymi drogi wpływająca zarówno na czas jak i koszt przejazdu[2].

Algorytmami pozwalającymi na uwzględnienie tych czynników są algorytm A*, algorytm Bellmana-Forda oraz algorytm Dijkstry.

Algorytm Dijkstry zaprojektowany został do znajdowania najkrótszej ścieżki w grafie o nieujemnych wagach krawędzi [3]. W wyniku wykonania algorytmu obliczone zostają odległości od węzła początkowego do każdego z węzłów w grafie. Wykonanie algorytmu do końca powoduje, że w każdym węźle grafu zostaje zapamiętana najmniejsza znaleziona odległość od węzła początkowego. W przypadku, jeżeli algorytm znajdzie krótszą drogę do węzła, który został już wcześniej przeanalizowany, to zamieni zapamiętane wartości wyznaczając nową krótszą ścieżkę. Algorytm Dijkstry jest używany do wyznaczania najkrótszej drogi oraz przy trasowaniu ruchu w sieciach komputerowych [3,6]. W tych zastosowaniach skrzyżowania przedstawiane są za pomocą węzłów grafu a łączące je drogi za pomocą krawędzi grafu. Każda krawędź opisana jest poprzez węzeł początkowy, węzeł końcowy oraz wagę, która odpowiada odległości pomiędzy krańcowymi punktami drogi. Dzięki przypisaniu wag krawędziom możliwe jest uwzględnienie długości dróg w obliczeniach. Każda krawędź grafu w algorytmie Dijkstry jest skierowana. Oznacza to, że może być przebyta tylko w jedną stronę. Jest to korzystne z tego względu, że ułatwia odwzorowanie dróg jednokierunkowym. Droga dwukierunkowa w takim układzie musi być reprezentowana przez dwie krawędzie – każda z nich będzie opisywała inny kierunek ruchu.

Algorytm Bellmana-Forda nie posiada ograniczenia w postaci konieczności definiowania krawędzi o nieujemnych wagach. Jest to dopuszczalne, jednak nie może wystąpić cykl osiągalny ze źródła posiadający łączną ujemną wagę. Inny sposób działania skutkuje jednak większą złożonością czasową algorytmu Bellmana-Forda. Nie wydaje się więc rozsądnym korzystanie z niego w zastosowaniach wyznaczania tras, gdyż koszt przejazdu nie może być ujemny co implikuje brak konieczności usuwania tego ograniczenia.

Dobrym algorytmem do wyznaczania tras jest algorytm A*. Jego działanie polega na wyznaczaniu najkrótszej trasy na podstawie kosztu już przebytej trasy oraz heurystycznego przewidywania kosztu trasy pozostałej do przebycia. Jednak jakość rozwiązań dostarczanych przez ten algorytm zależy od jakości funkcji heurystycznej i precyzji jej działania. Konieczne jest więc opracowanie dobrej funkcji heurystycznej co najczęściej stanowi poważne wyzwanie i jest trudne do osiągnięcia.

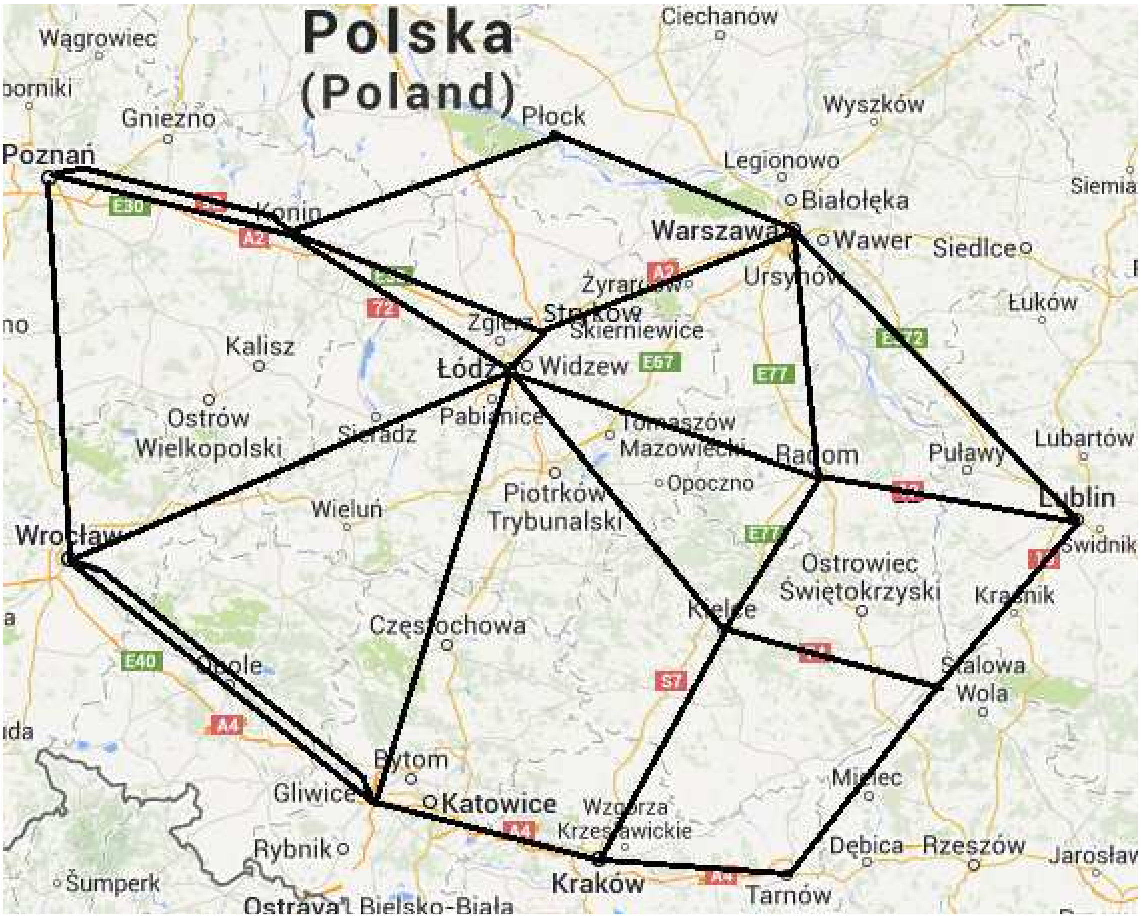
Ze względu na przedstawione powyżej cechy algorytm Dijkstry warty jest zastosowania w aplikacjach wyznaczania optymalnej trasy w znanej topologii dróg. W przypadku analizy złożonych problemów możliwe jest zastosowanie jego modyfikacji o zwiększonej szybkości działania [5] lub implementacji równoległych[4].

2. ULEPSZONY ALGORYTM DIJKSTRY

W celu spełnienia wymagań, które zostały sprecyzowane wcześniej w artykule, czyli umożliwienia wyszukiwania optymalnej trasy przy uwzględnieniu wielu parametrów jednocześnie w niniejszym artykule przedstawiona zostanie propozycja modyfikacji algorytmu Dijkstry. W algorytmie wprowadzono kilka zmian. Pierwszą z nich jest modyfikacja sposobu reprezentowania krawędzi grafu. Zamiast tradycyjnej krawędzi określanej przez węzeł początkowy, węzeł końcowy oraz koszt wprowadzono krawędź, która oprócz wymienionych cech posiada dodatkowo tablicę wag. Tablica ta pozwala na przechowywanie różnych wartości kosztu przebycia krawędzi. W rozpatrywanym przykładzie przeprowadzone zostały testy optymalizowania trasy z uwzględnieniem trzech różnych parametrów, których wartości zostały umieszczone w tablicy wag. Są to: długość odcinka drogi, średni czas przejazdu oraz koszt opłaty za przejazd drogą. Oczywiście wymieniona lista nie wyczerpuje wszystkich możliwości. Proponowana modyfikacja pozwala na zastosowanie dowolnie dużej liczby współczynników wagowych określających krawędź. Dodatkowo do opisu krawędzi dołączono jej nazwę. Dzięki temu możliwe jest przechowywanie nazwy drogi lub jej opisu.

Konieczne jest to ze względu na fakt, że pomiędzy dwoma wybranymi punktami często istnieje wiele alternatywnych dróg, które posiadają zupełnie inne parametry. Bez określenia nazwy drogi kierowca nie miał by możliwości zweryfikowania, którą z nich zaproponował algorytm.

Kolejną wprowadzoną modyfikacją jest dołączenie do algorytmu obliczającego koszt trasy wag z jakimi brane są pod uwagę poszczególne koszty przebycia krawędzi. Im wyższa waga zostanie przypisana do określonego parametru, tym większy wpływ ma on na wybór trasy. Przykładowo jeżeli do wymienionych wcześniej parametrów przypiszemy następujące wagi: odległość – waga 1, szybkość – waga 1, opłata za przebycie trasy – waga 5, to wówczas wysokość opłat ponoszonych za przejazd drogami będzie miała pięciokrotnie większy wpływ niż każdy z pozostałych dwóch parametrów. W celu przeanalizowania działania algorytmu opracowana została aplikacja implementująca jego działanie. Przygotowana została również struktura sieci połączeń drogowych pozwalająca na przetestowanie działania algorytmu. Struktura ta obejmuje kilkanaście ważnych miast lub węzłów drogowych położonych w południowej części Polski oraz połączenia drogowe pomiędzy nimi. Jej graficzną reprezentację przedstawiono na rysunku 1.



Rys. 1. Testowa struktura grafu odwzorowująca połączenia drogowe pomiędzy wybranymi miastami [źródło mapy: https://maps.google.pl]

Pomiędzy miastami przedstawionymi na powyższym grafie utworzono sieć połączeń. Niekiedy obejmuje ona kilka dróg pomiędzy dwoma tymi samymi punktami. Dodanie różnych połączeń pomiędzy miastami miało na celu umożliwienie dodania dróg posiadających różne parametry. Dla każdej z dróg opisano parametry zgodne z istniejącymi w rzeczywistości takie jak odległość oraz czas przejazdu. Dodatkowo dodano dodatkowy parametr jakim jest opłata za przejazd. Dla celów testowych założono, że wszystkie odcinki autostrad są płatne i przypisano do nich stawki. Lista połączeń wraz z ich parametrami została przedstawiona w tabeli 1. W strukturze grafu każde z wymienionych połączeń zostało przedstawione za pomocą dwóch krawędzi, z których każda reprezentuje inny kierunek ruchu.

Tab. 1. Lista połączeń w grafie testowym

Węzeł 1	Węzeł 2	odległość [km]	Czas przejazdu [min]	Opłata za przejazd [zł]	Nazwa drogi
Lublin	Warszawa	168	151	0	E372
Lublin	Radom	116	110	0	12
Warszawa	Radom	105	80	0	E77
Radom	Kielce	77	70	0	E77
Radom	Łódź	137	128	0	E48
Lublin	Annopol	77	74	0	19
Kielce	Annopol	96	84	0	74
Kielce	Łódź	146	132	0	74
Warszawa	Kielce	178	145	0	W2
Warszawa	Stryków	115	68	18	A2
Stryków	Łódź	22	27	0	E75
Warszawa	Łódź	138	123	0	DK72/E67
Stryków	Konin	111	62	30	A2
Łódź	Konin	112	109	0	DK72
Konin	Poznań	105	63	30	A2
Konin	Poznań	100	92	0	DK92
Annopol	Tarnów	136	131	0	DW984
Tarnów	Kraków	88	58	20	A2
Kraków	Gliwice	99	59	18	A4
Kielce	Kraków	117	110	0	S7
Kielce	Gliwice	188	154	0	E78
Łódź	Gliwice	230	163	0	E75
Gliwice	Wrocław	178	102	27	A4
Gliwice	Wrocław	173	167	0	DK94
Łódź	Wrocław	220	165	0	E67
Poznań	Wrocław	167	152	0	E261
Warszawa	Płock	116	94	0	E77
Konin	Płock	125	120	0	E92

Procedurę testową rozpoczęto od wyznaczenia w zaimplementowanej siatce połączeń trasy pomiędzy Tarnowem a Poznaniem. Wszystkie współczynniki zostały wzięte z wagą równą jeden, czyli miały jednakowy wpływ na wybór trasy. Dla uproszczenia przyjmiemy następujące oznaczenia współczynników: w_0 – waga odległości, w_1 – waga czasu przejazdu, w_2 – waga opłaty za przejazd. Koszt przebycia każdego odcinka trasy obliczany jest według wzoru: długość*w_1+czas przejazdu*w_2+opłata*w_3. Zastosowanie powyższych wartości współczynników spowodowało wybranie trasy wiodącej w przeważającej części autostradą A4, tak jak zaprezentowano to w pierwszej kolumnie wyników tabeli 2. Przyjmując wagi w_0=1,w_1=0,w_2=0 łatwo sprawdzimy, że jest to trasa najkrótsza. Jest to również trasa najszybsza co możemy stwierdzić stosując współczynniki w_0=0,w_1=1,w_2=0. Wyznaczona trasa obejmuje jednak odcinki płatne. Jeżeli istotne jest uniknięcie dodatkowych opłat za przejazd wystarczy odpowiednio zwiększyć wagę współczynnika w_2, aby algorytm zaczął unikać odcinków na których pobierane są opłaty. W przedstawionym przykładzie zwiększenie wartości tego współczynnika do 5 powoduje zmianę trasy, tak jak zaprezentowano to w tabeli 2. Warto zwrócić uwagę na fakt, że dopiero po przekroczeniu pewnej progowej wartości współczynnika nastąpiła zmiana trasy. Dzieje się tak dlatego, że koszt (*k*) każdej trasy kalkulowany jest zgodnie z wzorem 1:

$$k = \sum_{n=1}^{N}(odległość_n * w_0 + czas\_przejazdu_n * w_1 + opłata_n * w_2),\ n=1,2,\dots,N \qquad (1)$$

Gdzie:

N oznacza całkowitą liczbę odcinków trasy do punktu docelowego.

Trasa 1 posiada długość całkowitą d_1=532 km i łączny czas przejazdu t_1=371 minut. Opłaty jakie trzeba ponieść za przejechanie tej trasy wynoszą łącznie o_1=65zł. Długość trasy 2 wynosi d_2=590 km a czas jej przejazdu t_2=548. Nie trzeba jednak ponosić dodatkowych opłat za jej przejechanie – o_2=0. Trasy otrzymają jednakową ocenę jeżeli skalkulowane koszty ich przejechania będą jednakowe: $k_1 = k_2$. Jeżeli do tej równości wstawimy zależności z równania 1 otrzymamy:

$$d_1 * w_{01} + t_1 * w_{11} + o_1 * w_{21} = d_2 * w_{02} + t_2 * w_{12} + o_2 * w_{32} \quad (2)$$

gdzie:

w_{01}, w_{11},w_{21} – wagi przyjęte przy przeliczaniu trasy 1,

w_{02}, w_{12}, w_{22} – wagi przyjęte przy przeliczaniu trasy 2.

Aby określić przy jakiej wartości współczynnika w_{21} nastąpi zmiana trasy należy przekształcić równanie do postaci:

$$w_{21} = \frac{d_2 * w_{02} + t_2 * w_{12} + o_2 * w_{32} - d_1 * w_{01} - t_1 * w_{11}}{o_1} \quad (3)$$

Wstawiając wartości do równania 3 otrzymujemy w_{21}=3,62. Oznacza to, że przyjęcie większej wagi niż wyliczona spowoduje w przedstawionym przykładzie modyfikację przebiegu trasy. Jak łatwo zauważyć przy pomocy współczynników wagowych możemy łatwo ustalać proporcje jakie chcemy przyjąć pomiędzy poszczególnymi czynnikami określającymi koszt trasy. Z wzoru 3 jasno wynika, że współczynnik wagowy określa proporcję pomiędzy wartością parametru do którego jest przypisany a pozostałymi współczynnikami. Takie zorganizowanie problemu pozwala na łatwe i elastyczne określanie najdogodniejszej dla użytkownika opcji wyboru trasy. Zachowanie proporcji pomiędzy współczynnikami zgodne z równaniem 3 możemy łatwo sprawdzić wstawiając w przedstawionym przykładzie wyznaczania trasy wartości wag w_0=0,w_1=5,w_2=14. Otrzymamy wówczas w wyniku trasę 3. Jest ona zoptymalizowana zarówno pod względem kosztów jak i czasu przejazdu. Jak widać trasa obejmuje odcinki płatne, lecz odcinek pomiędzy Gliwicami a Wrocławiem pokonywany jest trasą o dłuższym czasie przejazdu lecz bezpłatną. Zachowanie takie wynika z położenia większego nacisku na minimalizację wysokości opłat drogowych (waga równa 14) niż na czas przejazdu (waga równa 5). Wybranie alternatywnej trasy omijającej wszystkie odcinki płatne znacznie wydłużyło by czas przejazdu co nie jest zgodne z założeniami określonymi przez współczynniki wagowe.

Tab. 2. Wyznaczone trasy przejazdu Tarnów – Poznań

		Trasa1	Trasa2	Trasa 3
Wartości współczyn-ników	w0	1	1	0
	w1	1	1	5
	w2	1	5	14
Wyznaczona trasa przejazdu		Tarnów do Poznań (968) Tarnów->Kraków (88)km trasą A2 ; Kraków->Gliwice (99)km trasą A4 ; Gliwice->Wrocław (178)km trasą A4 ; Wrocław->Poznań (167)km trasą E261 ;	Tarnów do Poznań (1138) Tarnów->Annopol (136)km trasą DW984 ; Annopol->Kielce (96)km trasą 74 ; Kielce->Łódź (146)km trasą 74 ; Łódź->Konin (112)km trasą DK72 ; Konin->Poznań (100)km trasą DK92 ;	Tarnów do Poznań (2712) Tarnów->Kraków (88)km trasą A2 ; Kraków->Gliwice (99)km trasą A4 ; Gliwice->Wrocław (173)km trasą DK94 ; Wrocław->Poznań (167)km trasą E261 ;

WNIOSKI

Określanie optymalnej trasy jest w dzisiejszych czasach konieczne w wielu dziedzinach życia. Istotna jest również elastyczność oferowanego rozwiązania pozwalająca użytkownikowi na samodzielne definiowanie parametrów, które wpływają na wybór trasy. Nie mniej istotnym aspektem jest zapewnienie możliwości przypisania wag, z jakimi poszczególne parametry będą uwzględniane przy wyborze optymalnej trasy. Zaproponowana w artykule modyfikacja algorytmu Dijkstry pozwala na uwzględnianie dowolnej liczby parametrów podczas wyboru drogi, dzięki przypisaniu wielu niezależnych wag do każdej z krawędzi grafu odwzorowującego połączenia drogowe. Umożliwia to elastyczne przeliczanie tras, dowolny wybór parametrów uwzględnianych przy wyborze drogi oraz określanie w jakim stopniu będą miały one wpływ na ostateczny wynik. Osiągane jest to poprzez uwzględnienie każdej z wag przypisanych do krawędzi grafu a następnie przemnożenie ich przez przypisane do nich wagi określające wpływ jaki poszczególne współczynniki powinny mieć na ostateczny wybór trasy. Dzięki takiemu podejściu każdy użytkownik może samodzielnie zdefiniować optymalne dla niego czynniki wpływające na wybór trasy i określić ich stopień istotności.

Streszczenie

Wyznaczenie optymalnej trasy przejazdu ma coraz większe znaczenie w logistyce. Ważne są przede wszystkim koszty transportu oraz jego czas. Istotne jest więc zaplanowanie trasy w taki sposób by zoptymalizować żądane czynniki. Możliwe jest to dzięki zastosowaniu algorytmu pozwalającego uwzględnić koszt trasy. Należy jednak zwrócić uwagę na fakt, że na koszt przebycia trasy może składać się wiele czynników takich jak odległość, czas przejazdu, opłaty za przejazd określonymi drogami oraz inne. Konieczne jest więc stosowanie algorytmów pozwalających na znalezienie optymalnej trasy, przy uwzględnieniu wszystkich wymaganych czynników. Zaproponowana w artykule modyfikacja algorytmu Dijkstry pozwala na uwzględnianie dowolnej liczby parametrów podczas wyboru drogi, dzięki przypisaniu wielu niezależnych wag do każdej z krawędzi grafu reprezentującego sieć połączeń drogowych. Umożliwia to elastyczne przeliczanie tras, dowolny wybór parametrów uwzględnianych przy wyborze drogi oraz określanie w jakim stopniu będą miały one wpływ na ostateczny wynik. Osiągane jest to poprzez uwzględnienie każdej z wag przypisanych do krawędzi grafu a następnie przemnożenie ich przez przypisane do nich wagi określające wpływ jaki poszczególne współczynniki powinny mieć na ostateczny wybór trasy. Dzięki takiemu podejściu każdy użytkownik może samodzielnie zdefiniować optymalne dla niego czynniki wpływające na wybór trasy i określić ich stopień istotności.

Autor publikacji jest uczestnikiem projektu "Kwalifikacje dla rynku pracy - Politechnika Lubelska przyjazna dla pracodawcy" współfinansowanego przez Unię Europejską w ramach Europejskiego Funduszu Społecznego.

Algorithms for optimal route determine

Abstract

The importance of optimal route determination is very high today, especially in logistics. The most important aspects are the transport costs and its time. It is therefore essential to plan a route in such a way that allows to optimize the desired factors. It is possible by usage of an algorithm allowing for taking into account the route cost. It should be noted that the cost of traveling the route may consist of a number of factors such as distance, travel time, road fees and other. It is necessary to use algorithms that allow for finding optimal route by taking into account all the demanded factors. The proposed modification of Dijkstra's algorithm allows for taking into account any number of parameters when selecting the route. It is possible thanks to assigning a number of independent weights to each edge of the graph representing the network of roads. This allows for flexible route calculation and gives a possibility of each necessary parameter choice to take it into account when calculating a path and determining their impact on the final result. This is achieved by considering all weights assigned to the edges of the graph and then multiplying them by weights assigned to them. This operation makes it possible to determine the impact that each factor should have on the final choice of route. This approach allows user for defining the optimal factors influencing the choice of routes and determine their significance.

BIBLIOGRAFIA

1. Chabini I., Discrete Dynamic Shortest Path Problems In Transportation Applications: Complexity And Algorithms With Optimal Run Time. Transportation Research Records , 1645:170–175, 1998
2. Colombo R. M., P. Goatin, and M. Rosini. On the modeling and management of traffic. Quaderni del Seminario Matematico di Brescia , 14, 2010

3. DongKai Fan ; Ping Shi , Improvement of Dijkstra's algorithm and its application in route planning , Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference, s. 1901 - 1904
4. Jasika N., et. al.,Dijkstra's shortest path algorithm serial and parallel execution performance analysis, MIPRO, 2012 Proceedings of the 35th International Convention , s. 1811 - 1815
5. Nikos Anastopoulos, Konstantinos Nikas, Georgios Goumas and Nectarios Koziris, „Employing Transactional Memory and Helper Threads to Speedup Dijkstra’s Algorithm“ ,2009
6. Shu Yang ; Chunhua Li, An enhanced routing method with Dijkstra algorithm and AHP analysis in GIS-based emergency plan , Geoinformatics, 2010 18th International Conference, s. 1-6
7. Wilson R.J., Wprowadzenie do teorii grafów, PWN, Warszawa 2000