

Application of graph algorithms for optimal path planning of mobile robots

Zastosowanie algorytmów grafowych do planowania optymalnej trasy przejazdu robotów mobilnych

JAKUB GURGUL
ANDRZEJ JAŁOWIECKI*

DOI: <https://doi.org/10.17814/mechanik.2025.02.2>

The objective of the research described in this article was to develop a model for planning the optimal traversal route for a wheeled mobile robot based on a Digital Terrain Model (DTM), utilizing graph algorithms. A methodology was developed to transform the DTM into a graph representation, enabling the efficient application of graph algorithms for route determination. Three algorithms were implemented in Python: Prim's algorithm, Dijkstra's algorithm, and the A* algorithm. These algorithms were applied to graph-based terrain analysis, comparing their performance in terms of route feasibility, path length, and computation time. Notably, the A* algorithm yielded the most promising computational efficiency results in the conducted tests.

KEYWORDS: mobile robotics, graph algorithms, path planning, global navigation, digital terrain model

Celem prac opisanych w niniejszym artykule było opracowanie modelu planowania optymalnej trasy przejazdu dla kołowego robota mobilnego na podstawie Numerycznego Modelu Terenu (NMT), z wykorzystaniem algorytmów grafowych. Opracowano metodykę pozwalającą na sprowadzenie NMT do postaci grafowej, dzięki czemu możliwe było efektywne zastosowanie algorytmów grafowych do wyznaczenia trasy. Zaimplementowano trzy algorytmy w języku Python: algorytm Prima, Dijkstry i A*. Zastosowano je do analizy grafowej terenu, porównując wyniki m.in. pod kątem zdolności do wyznaczenia trasy, długości trasy i czasu obliczeń. W przeprowadzonych testach algorytm A* uzyskał najbardziej obiecujące rezultaty czasowe.

SŁOWA KLUCZOWE: robotyka mobilna, algorytmy grafowe, planowanie trasy, nawigacja globalna, numeryczny model terenu

Wprowadzenie

Aktualnie kładzie się bardzo duży nacisk na to, aby robot był w stanie pracować w całkowicie autonomiczny sposób lub ograniczający do minimum interakcję z operatorem. Jednym z podstawowych zadań, jakie platforma robotyczna powinna wykonywać samodzielnie, jest przemieszczenie się. Korzystając z wbudowanych systemów wizyjnych oraz algorytmów przetwarzania obrazu, roboty mobilne są w stanie planować ścieżkę przejazdu w czasie rzeczywistym podczas pracy w nieznanym dynamicznym środowisku [1, 2].

Przykłady takiego inteligentnego planowania przemieszczenia możemy znaleźć w autonomicznych samochodach [3] lub maszynach rolniczych. Zdolność do samodzielnego przemieszczania jest szeroko używana

w wielu sytuacjach: eksploracja innych planet [4], eksploracja niebezpiecznego otoczenia, takiego jak strefy napromieniowania [5], czy obszary klęsk naturalnych [6]. Dzięki możliwości autonomicznego przemieszczania się oraz pracy w szczególnych warunkach, można przyspieszyć monotonne prace oraz zdecydowanie ograniczyć ryzyko, jakie w niebezpiecznych sytuacjach ponoszą ludzcy operatorzy.

Podczas przeprowadzania procesu autonomicznego przemieszczania się możemy wyszczególnić dwa główne obszary: określenie przejezdności przez dany teren wraz z wyznaczeniem obszarów dostępu (rys. 1a, rys. 1b) oraz oznaczenie punktów kontrolnych wraz z wyznaczeniem pomiędzy nimi trasy przejazdu (rys. 1c, rys. 1d).

W pierwszym etapie trudność związana jest z określeniem kosztu poruszania się po danym terenie. Koszt ten zazwyczaj odnosi się do bezpieczeństwa poruszania się robota, czasu, jaki pochłonie, oraz wydatków energetycznych niezbędnych do zrealizowania przejazdu. Dodatkowym skutkiem określania przejezdności jest poinformowanie operatora o tym, że niektóre obszary są całkowicie niedostępne dla robota lub że ich odwiedzenie może być niezwykle niebezpieczne.



(a) Aerial photograph of the area
Zdjęcie obszaru wykonane z powietrza



(b) Identification of accessible area
Wyznaczenie obszaru dostępności



(c) Marking of control points
Oznaczenie punktów kontrolnych



(d) Determination of the travel path
Wyznaczenie ścieżki przejazdu

Fig. 1. Visualization of the route planning process
Rys. 1. Wizualizacja procesu wyznaczania ścieżki przejazdu

* Inż. Jakub Gurgul – jakub.gurgul@student.polsl.pl – Wydział Matematyki Stosowanej, Politechnika Śląska
Dr inż. Andrzej Jałowiecki – andrzej.jalowiecki@polsl.pl – Wydział Mechaniczny Technologiczny, Politechnika Śląska

W drugim obszarze największą trudnością jest wyznaczenie takiej ścieżki poruszania się, która będzie kompromisem pomiędzy bezpieczeństwem, czasem oraz wydatkami energetycznymi. Jest on rozwiązywany z wykorzystaniem algorytmów do planowania ścieżki przejazdu. Algorytmy te pozwalają na opracowanie możliwych ścieżek na podstawie wprowadzonych informacji dotyczących obszaru oraz wybranych punktów kontrolnych.

Celem prac opisanych w artykule było stworzenie modelu do planowania optymalnej trasy przejazdu na podstawie Numerycznego Modelu Terenu (NMT) z wykorzystaniem algorytmów grafowych. Model jako dane wejściowe przyjmował NMT, pozyskany np. w procesie fotogrametrii wykonanej na podstawie zdjęć z drona i współrzędne położenia robota mobilnego oraz jego punktu docelowego. Jest to symulacja wykonywania zadania przemieszczenia się w grupie robotów. Pierwszym etapem jest przelot dronem zwiadowczym i wykonanie zdjęć zadanego obszaru. Drugi etap to właściwa operacja przemieszczenia wykonywana przez lądowego robota mobilnego. Taka współpraca pomiędzy robotami jest coraz bardziej popularna; przykładem może być łaźnik Perseverance oraz wielowirnikowiec Ingenuity, które współpracują aktualnie na powierzchni Marsa [7].

Ze względu na złożoność zagadnienia, w niniejszej pracy został opisany tylko sposób oceny przejeżdżalności oraz zaplanowania trasy dla robota mobilnego w ujęciu globalnym na otwartym terenie w otoczeniu niewykorzystującym tendencji do dynamicznych zmian w czasie.

Numeryczny Model Terenu

Numeryczny Model Terenu (NMT), znany także jako Numeryczny Model Rzeźby Terenu (NMRT), to reprezentacja trójwymiarowej struktury terenu za pomocą danych liczbowych. Jest to sposób reprezentacji terenu szeroko stosowany w inżynierii, geoinformatyce, geologii, hydrologii i wielu innych dziedzinach, gdzie potrzebne są dokładne informacje o topografii terenu [8]. Najczęściej spotykamy dwa podstawowe rodzaje NMT, które są oparte na różnych strukturach geometrycznych: *Grid* – model w postaci regularnej siatki kwadratów oraz *Tin* – model w postaci nieregularnej siatki trójkątów.

Struktura modelu oparta jest na punktach terenowych, które są jednostkami reprezentującymi powierzchnię. Dla każdego punktu przechowywane są trzy współrzędne: X oraz Y określają położenie na płaszczyźnie, a Z określa wysokość danego punktu nad poziomem morza. Kolejnym ważnym elementem jest rozdzielczość modelu, która definiuje odległość między sąsiadującymi punktami terenowymi. Wysoka rozdzielczość oznacza, że model zawiera więcej punktów, co umożliwia bardziej szczegółowe odwzorowanie topografii terenu.

Planowanie trasy przejazdu

Samo pojęcie planowania przejazdu jest niezwykle szerokie [9], ponieważ proces ten jest zależny od parametrów, takich jak typ platformy robotycznej czy środowisko i specyfika operacji. Innym zasadom pracy podlegają jednostki lądowe, powietrzne lub wodne.

Charakterystyka ich pracy oraz możliwości poruszania się muszą być wzięte pod uwagę podczas wyboru oraz opracowywania systemu planowania trasy. Kolejnym ważnym aspektem jest podział środowisk operacji ze względu na otwartość, posiadaną wiedzę początkową oraz dynamiczność środowiska. Pierwsze kryterium odnosi się do tego, czy mamy do czynienia z poruszaniem się po otwartym terenie (polana, parking itp.), czy poruszaniem się wewnątrz budynków, czy w innych zamkniętych strukturach.

Drugie kryterium dotyczy wiedzy, jaką posiadamy na starcie i możemy nią zasilić algorytmy dodatkowymi danymi. Takimi dodatkowymi informacjami mogą być zdjęcia wykonane z powietrza, plany budynku, dane zebrane z czujników w trakcie wcześniejszych przejazdów lub wiedza ekspercka operatora.

Ostatnie kryterium określa zmienność środowiska oraz tendencję do tych zmian. Może ono określać potencjalne obszary, które w czasie zmieniają swoje parametry, wskutek czego zmienia się ich przejeżdżalność. Są to wszelkiego rodzaju ruchome przeszkody, luźne podłoże (np. piasek czy żwir) oraz innego rodzaju sytuacje losowe (np. pożar lub zawalenie się drogi). Wszystkie te czynniki mają istotny wpływ na zaplanowanie trasy i jej dokładne zrealizowanie.

Proces planowania trasy jest sposobem przetworzenia pierwotnego opisu otoczenia na inną formę, bardziej dogodną dla przyjętego sposobu rozwiązywania problemu. Kolejnym krokiem jest znalezienie takich przekształceń, które pozwolą przenieść układ początkowy do zadanego stanu końcowego. W ujęciu robotyki mobilnej jest to takie wyznaczenie krzywej geometrycznej, która pozwoli na przejazd pomiędzy początkową pozycją robota a zadaną pozycją końcową. Dodatkowymi wymaganiami dotyczącymi ścieżki jest, aby robot nie wchodził w kolizję z innymi obiektami otoczenia oraz aby ścieżka była możliwa do wykonania przez platformę w rzeczywistości. Drugie założenie jest szczególnie związane z konstrukcją robota (napęd, wymiary fizyczne itp.) oraz jego dynamiką (zakres ruchów możliwych do wykonania).

Algorytmy planowania ścieżki przejazdu możemy generalnie podzielić ze względu na dostępność informacji o otoczeniu na metody globalne oraz metody lokalne. W artykule poświęcono uwagę tylko pierwszej grupie metod.

Metody globalnej nawigacji

Zakładają obliczenie całej ścieżki w sposób iteracyjny. Wykorzystanie takich metod zakłada również z góry znajomość całego otoczenia (rozkład pomieszczeń, pozycję robota, informacje o przeszkodach itp.). Największą zaletą jest otrzymanie w efekcie optymalnej ścieżki przejazdu, wadą natomiast – duża złożoność obliczeniowa oraz mała odporność na zmiany w środowisku. Zazwyczaj w praktyce nie jest możliwe tak dokładne uzyskanie informacji o otoczeniu, w związku z czym te metody są zazwyczaj wykorzystywane do wstępnego określenia ruchu. Dlatego są również nazywane metodami *offline*, ponieważ są wstępnym planem ruchu, który nie jest aktualizowany na bieżąco nowymi danymi.

W metodach globalnych możemy zasadniczo wyróżnić trzy strategie tworzenia trasy przejazdu.

1. Plany przejazdu to podejście oparte na geometrycznym sposobie rozwiązywania problemu planowania ścieżki. Strategia ta zakłada wyrażenie przestrzeni otaczającej robota w postaci grafu z siecią możliwych połączeń. Jako przykłady algorytmów z tej grupy możemy podać grafy widoczności [10] czy sieci podcelów [11].

2. Dekompozycja komórek zakłada podział całej przestrzeni na zestaw prostszych podkomórek w celu obliczenia optymalnej ścieżki poruszania się pomiędzy nimi. Do tej grupy możemy zaliczyć dekompozycję trapezową [12] czy metodę maksymalnych obszarów wypukłych [13].

3. Pola potencjałów, elementy przestrzeni (robot, przeszkody) modelowane są jako wzajemnie oddziałujące na siebie ładunki elektryczne. Kierunek ruchu robota jest wypadkową działających na niego sił, przeszkody generują siły odpychające, natomiast cel ruchu generuje siły przyciągające.

Proces przetwarzania siatki terenu na graf

Pierwszym zadaniem przed przystąpieniem do tworzenia modelu było zaprojektowanie procesu przetwarzania otrzymanego Numerycznego Modelu Terenu na postać grafową. NMT zostaje dostarczony w formie pliku z rozszerzeniem VTK (ang. *Visualization Toolkit*). Plik VTK to plik tekstowy lub binarny zawierający dane potrzebne do wizualizacji trójwymiarowej sceny, takie jak informacje o geometrii, kolorach, teksturach, oświetleniu i innych właściwościach obiektów. Struktura pliku VTK obejmuje nagłówki opisujące typ danych, liczbę punktów, komórek, właściwości i innych elementów niezbędnych do odtworzenia trójwymiarowej sceny.

Przed przystąpieniem do procesu przetwarzania, przyjęto poniższe zasady.

- Wierzchołkami w grafie zostaną konkretne punkty w siatce terenu. Do ich identyfikacji posłuży unikatowy numer przyznawany od punktu z koordynatami (0,0) i następnie dla każdego kolejnego punktu w kolumnie. Po skończeniu punktów w danej kolumnie przechodzimy do kolejnej itd. aż do przepisania wszystkich punktów w siatce.
- Pojedynczy wierzchołek ma krawędzie tylko z wierzchołkami odpowiadającymi sąsiadującym punktom w siatce. Z każdego wierzchołka możemy przejść maksymalnie do 4 sąsiadujących punktów (góra, dół, lewo, prawo).
- Ponieważ graf będzie używany w procesie planowania ścieżki przejazdu, krawędziom zostały przypisane odpowiednie wagi. Są one absolutną różnicą znormalizowanej wartości elewacji pomiędzy sąsiadującymi wierzchołkami.

Zbiór danych testowych

Podczas prac opisanych w niniejszym artykule pliki VTK zostały wygenerowane w sposób pseudolosowy z wykorzystaniem algorytmu generowania szumu Perlina. Tworzone tereny mają rozmiar 1000×1000

punktów, a wartość elewacji mieści się w zakresie od 0 do 2 metrów.

Po odczytaniu informacji zapisanych w pliku został przeprowadzony proces normalizacji danych dotyczących elewacji. Ta operacja była konieczna do przeprowadzenia ze względu na późniejsze operacje służące przekształceniu terenu na postać grafową.

W ramach testu przygotowano zestaw 1000 terenów. Każdy składa się z siatki o wymiarach 1000×1000 punktów, co odpowiada w rzeczywistości obszarowi o powierzchni około 10×10 metrów, przy zagęszczeniu 100 punktów na metr. Poziom elewacji na tym obszarze waha się od 0 do 2 metrów, co pozwala symulować zróżnicowane warunki terenowe.

Podczas procedury przetwarzania modeli terenowych na postać grafową punkt startowy oraz końcowy zostały wybrane losowo z zastrzeżeniem, że odległość między punktem startowym a końcowym musi wynosić co najmniej 5000 punktów w przestrzeni 2D, zgodnie z metryką euklidesową. Ten warunek ma na celu wprowadzenie zróżnicowania i realistyczności do testów, imitując różnorodne warunki terenowe oraz długości tras, jakie mogą wystąpić w rzeczywistości. Dodatkowo, ze względu na warunek minimalnej długości, wymusza na testowanych algorytmach przeprowadzenie bardziej skomplikowanych obliczeń.

Kryteria porównawcze

Wyznaczenie trasy do wierzchołka końcowego stanowi podstawowe kryterium porównawcze dla algorytmów grafowych. Algorytmy muszą być w stanie skutecznie odnaleźć trasę prowadzącą od punktu startowego do docelowego wierzchołka w grafie, co jest fundamentalnym aspektem ich funkcjonalności.

Czas prowadzenia obliczeń w przypadku robotyki mobilnej jest kluczowy ze względu na potrzebę sprawnego planowania przemieszczenia w przestrzeni. Czas wykonywania algorytmów będzie mierzony w sekundach.

Długość trasy – preferowane są krótsze trasy, ponieważ pozwalają na bardziej efektywne i oszczędne wykorzystanie zasobów robota. W celu poprawnego zmierzenia odległości użyto metryki euklidesowej dla przestrzeni 3D.

Specyfikacja platformy testowej

W ramach testów wykorzystano dwie platformy: komputer stacjonarny klasy PC (ang. *Personal Computer*) oraz komputer jednopłytkowy typu SBC (ang. *Single Board Computer*). Platformy te zostały wybrane w celu uzyskania porównania między wydajnością obliczeń prowadzonych na sprzęcie o różnej specyfikacji, co pozwala na ocenę przenośności algorytmów na środowiska z ograniczonymi zasobami obliczeniowymi, takimi jak roboty mobilne.

Wykorzystane algorytmy grafowe

Do implementacji, zostały wybrane trzy algorytmy pozwalające na wyznaczenie najkrótszej ścieżki w grafie

ważonym z wierzchołka startowego do wskazanego. Każdy z algorytmów na tym etapie został poddany wstępnemu testowi. Zadaniem było wyznaczenie ścieżki przejazdu z punktu (0,0) do środka wygenerowanego terenu. Każdy z algorytmów miał wykonać to zadanie na tym samym zbiorze trzech wygenerowanych terenów, a rozmiar każdego z nich to 1000×1000 punktów.

Algorytm Prima jest używany do znajdowania minimalnego drzewa rozpinającego w grafie ważonym. Schemat działania algorytmu Prima opiera się na stopniowym dodawaniu do drzewa rozpinającego krawędzi o najmniejszych wagach, zaczynając od dowolnego wierzchołka. Niestety ten algorytm nie był w stanie wyznaczyć ścieżki przejazdu (obliczenia były prowadzone ponad trzydzieści minut) w testowym terenie, więc został wykluczony z udziału w dalszych testach wydajnościowych.

Algorytm Dijkstry znajduje najkrótsze ścieżki między wierzchołkiem startowym a wszystkimi innymi wierzchołkami w grafie ważonym. Jego głównym celem jest minimalizacja sumy wag krawędzi w trakcie poruszania się od wierzchołka startowego do pozostałych wierzchołków.

Algorytm A* jako jedyny z wybranych jest algorytmem heurystycznym, służącym, tak jak poprzednie, do przeszukiwania grafu, który efektywnie znajduje najkrótszą ścieżkę między dwoma wierzchołkami w grafie ważonym. Wykorzystuje on funkcję, która szacuje koszt przemieszczenia od danego wierzchołka do wierzchołka końcowego.

Otrzymane wyniki

W tabeli zostały przedstawione wyniki działania algorytmów. W tabeli nie zostały ujęte żadne wartości z kryterium długości wyznaczonej trasy. Jest to spowodowane tożsamyimi wartościami odległości tras wyznaczonych przez obydwa algorytmy na obydwu platformach testowych.

TABLE. Comparison of algorithm performance results
TABELA. Porównanie wyników działania algorytmów

Kryterium porównawcze	Algorytm Dijkstry		Algorytm A*	
	PC	SBC	PC	SBC
Wyznaczanie trasy (%)	100	100	100	100
Średni czas obliczeń (s)	3,57 ± 0,2	0,06 ± 0,01	4,20 ± 0,2	0,09 ± 0,01

Podsumowanie

We wszystkich przypadkach testowych oba algorytmy na obydwu platformach były w stanie wyznaczyć trasę prowadzącą od wybranego punktu startowego do określonego punktu końcowego. Uzyskanie poprawnych ścieżek w każdym przypadku podkreśla zdolność algorytmów do wyznaczania optymalnych tras, przy uwzględnieniu struktury grafu oraz różnych wartości wag krawędzi. Na podstawie tego kryterium, ze względu na otrzymane wyniki, nie było możliwe określenie jednoznacznie lepszego algorytmu.

Uzyskane wyniki czasu obliczeń dla algorytmów na różnych platformach wskazują na znaczące różnice

w efektywności obu metod. Na komputerze PC średni czas obliczeń dla algorytmu A* wyniósł 0,0601 sekundy, podczas gdy dla Dijkstry był to znacznie dłuższy czas, średnio 3,5764 sekundy. Analogicznie, na komputerze SBC algorytm A* osiągnął krótszy czas obliczeń wynoszący 0,0934 sekundy, podczas gdy Dijkstrze wyznaczenie trasy zajęło średnio ponad 4 sekundy. Na podstawie kryterium czasu obliczeń znacząco więc wygrywa algorytm A*.

We wszystkich przypadkach testowych oba algorytmy wyznaczyły dokładnie tę samą trasę jako najlepsze rozwiązanie. Ten wynik jest szczególnie istotny, gdyż wskazuje na jednomyślność algorytmów w wyborze optymalnej drogi. W przypadku tego kryterium, różnice w podejściach obu algorytmów do rozwiązywania problemu nie miały wpływu na wyniki, co wskazuje na jednolitość w procesie wyznaczania najbardziej efektywnej drogi.

Wnioski z przeprowadzonych badań sugerują, że algorytm A* stanowi optymalne rozwiązanie w kontekście czasu obliczeń, co jest kluczowe w praktycznych zastosowaniach. Dalsze prace autorów koncentrują się na rozbudowie modelu do postaci pełnoprawnego systemu nawigacyjnego. Dodatkowo planowana jest implementacja algorytmów z grupy metod lokalnych, umożliwiających dynamiczną nawigację w zmieniającym się środowisku oraz uzupełnienie brakujących informacji na bieżąco, w trakcie trwania ruchu.

LITERATURA

- [1] Jiang J., Xin J. „Path planning of a mobile robot in a free-space environment using q-learning”. *Progress in Artificial Intelligence*. Vol. 8, No. 1 (2019): 133–142.
- [2] Mohanan M.G., Salgoankar A. „A survey of robotic motion planning in dynamic environments”. *Robotics and Autonomous Systems*. Vol. 100, No. 1 (2018): 171–185.
- [3] Shreyas V., Bharadwaj S.N., Srinidhi S., Ankith K.U., Rajendra A.B. „Self-driving cars: An overview of various autonomous driving systems”. *Advances in Data and Information Sciences: Proceedings of IC-DIS 2019*. Indie: Springer, 2020.
- [4] Williford K.H., Farley K.A., Stack K.M., Allwood A.C., Beaty D., Beegle L.W., Bhartia R., Brown A.J., de la Torre Juarez M., Hamran S.-E. i in. „The NASA Mars 2020 Rover Mission and the Search for Extraterrestrial Life”. *From Habitability to Life on Mars*. Amsterdam: Elsevier, 2018.
- [5] Groves K., Hernandez E., West A., Wright T., Lennox B. „Robotic exploration of an unknown nuclear environment using radiation informed autonomous navigation”. *Robotics*. Vol. 10, No. 2 (2021): 78.
- [6] Wu M. „Robotics applications in natural hazards”. *Highlights in Science, Engineering and Technology*. Vol. 43, No. 1 (2023): 273–279.
- [7] Sasaki T., Otsu K., Thakker R., Haesaert S., Aghamohammadi A.-A. „Where to map? Iterative rover-copter path planning for Mars exploration”. *IEEE Robotics and Automation Letters*. Vol. 5, No. 2 (2020): 2123–2130.
- [8] Numeryczny model terenu (NMT). <https://www.geoportal.gov.pl/dane/numeryczny-model-terenu-nmt/> (dostęp: 10 marca 2025).
- [9] Tzafestas S.G. „Mobile Robot Control and Navigation: A Global Overview”. *Journal of Intelligent & Robotic Systems*. Vol. 91, No. 1 (2018): 35–58.
- [10] Lazarowska A. „Zastosowanie grafu widoczności w planowaniu trasy przejścia statku”. *Scientific Journal of Gdynia Maritime University*. R. 98 (2017): 116–121.
- [11] Lee G.T., Kim K. „A controllable agent by subgoals in path planning using goal-conditioned reinforcement learning”. *IEEE Access*. Vol. 11, No. 1 (2023): 33812–33825.
- [12] Eapen N.A. „Path planning of a mobile robot among curved obstacles through tangent drawing and trapezoidal decomposition”. *Engineering Science and Technology, an International Journal*. Vol. 24, No. 6 (2021): 1415–1427.
- [13] Vasquez-Gomez J.I., Marciano-Melchor M., Valentin L., Herrera-Lozada J.C. „Coverage path planning for 2D convex regions”. *Journal of Intelligent & Robotic Systems*. Vol. 97, No. 1 (2020): 81–94. ■