UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Programowanie Robotów Mobilnych

Planowanie ścieżki globalnej

Rafał Szczepański
e-mail: szczepi (at) umk.pl
www.umk.pl/~szczepi

22.05.2023

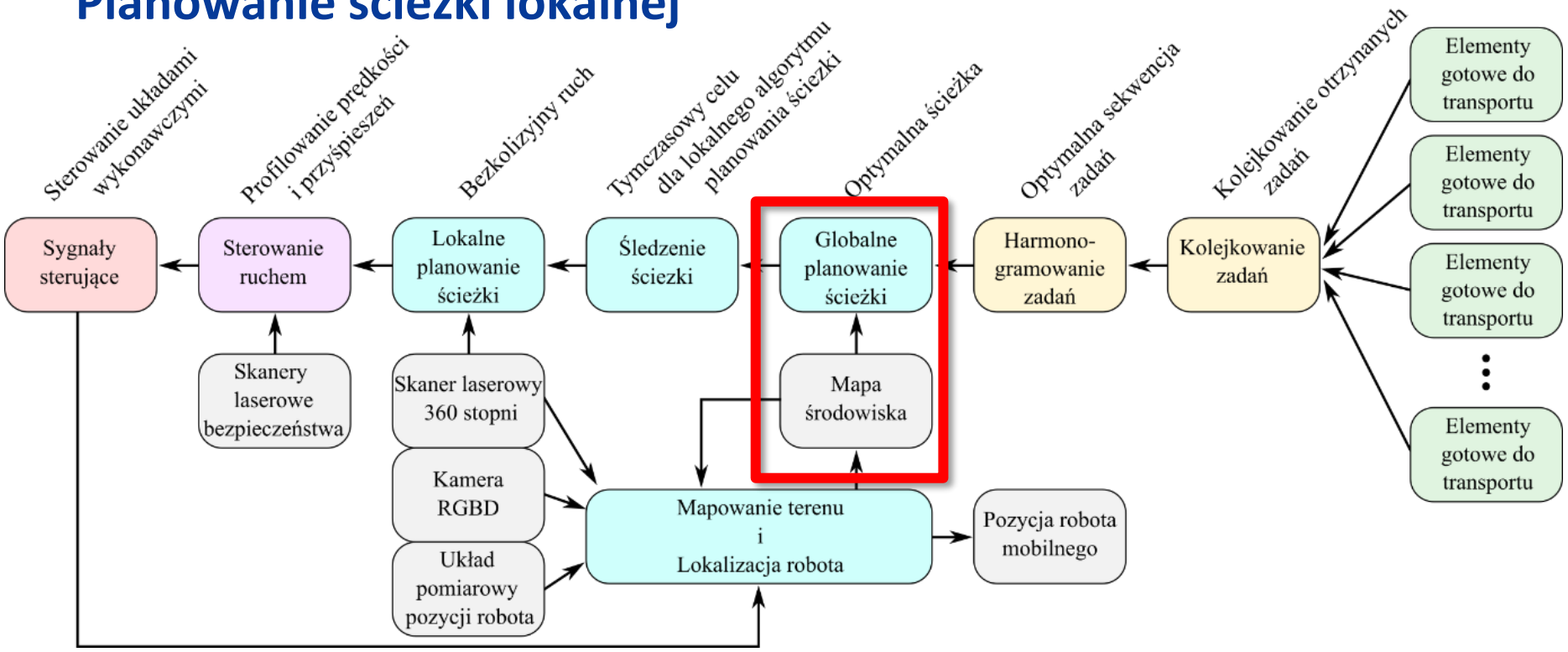
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Plan prezentacji

- Algorytm Dijkstra
- Algorytm A*
- Algorytm eksplorowania drzew losowych
- Algorytm mrówkowy

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Planowanie ścieżki lokalnej



Założenia:

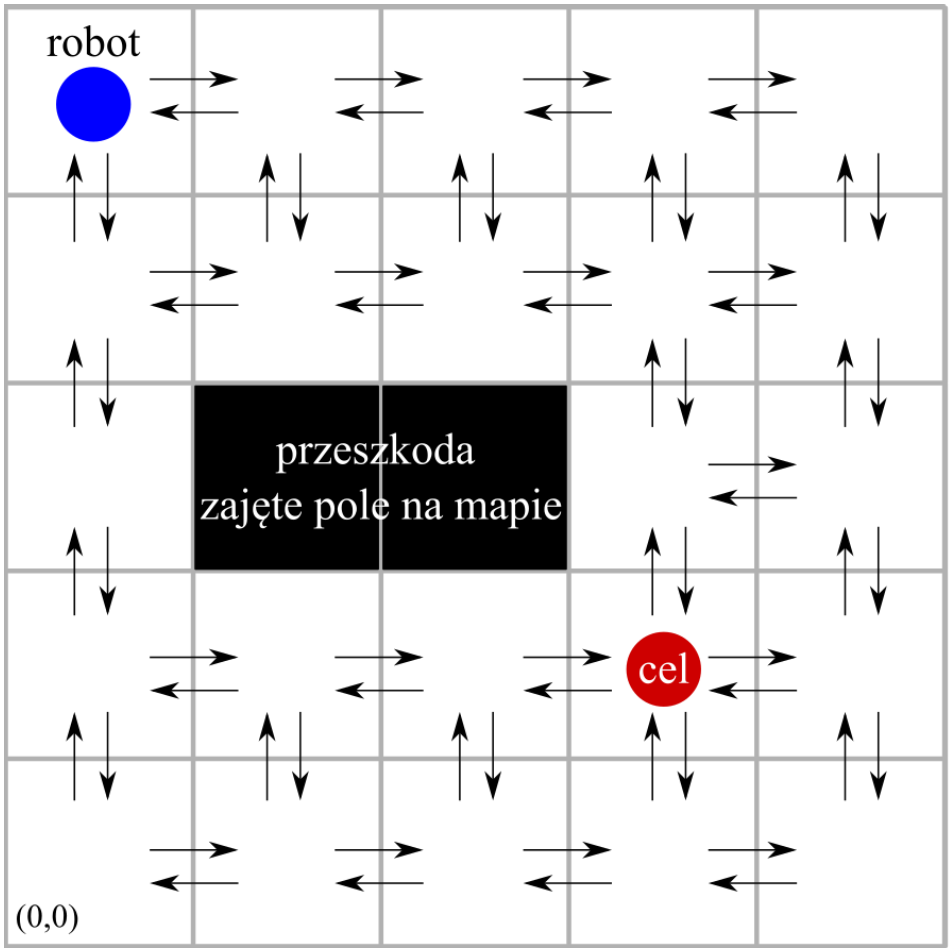
- Znajomość pozycji robota
- Znajomość pozycji zadanej (cel)
- Znajomość mapy środowiska

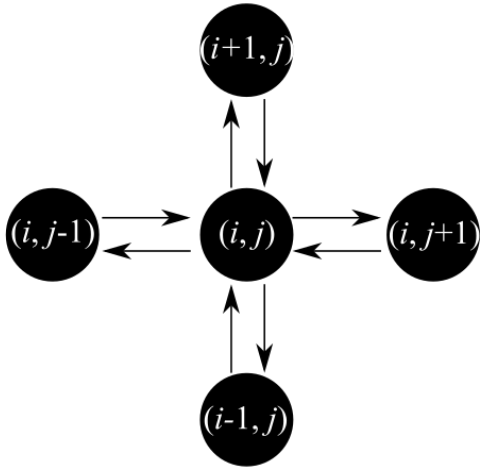
Oczekiwania:

- Znalezienie optymalnej ściezki

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Postać grafowa mapy





UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Algorytm Dijkstra

Algorytm znajdujący najkrótszą ścieżkę z pojedynczego źródła w grafie o nieujemnych wagach krawędzi. Algorytm bazuje na przeszukiwaniu drzewa wszerz (ang. Breadth First Search, BFS).

Algorytm rozpoczynając od punktu startowego przeszukuje drzewo w szerz aktualizując odległość od źródła do każdego z wierzchołków.

Wykonując kolejne kroki wybierając kolejne wierzchołki, do których dojście z wierzchołka początkowego jest wykonywane najmniejszym kosztem algorytm przeszukuje drzewo aż dojdzie do wierzchołka docelowego

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

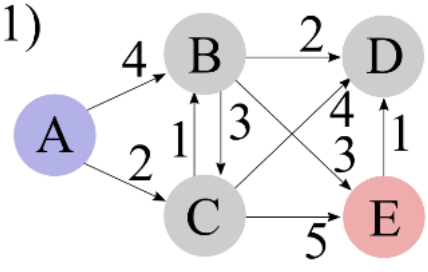
Algorytm Dijkstra

	Algorytm: Pseudokod algorytmu Dijkstra
	Wejście: G , s (wierzchołek początkowy), d (wierzchołek docelowy) Wyjście: ścieżka do celu
1	Dla każdego wierzchołka v w grafie G
2	$d[v] \leftarrow \infty$
3	$\text{poprzednik}[v] \leftarrow$ niezdefiniowano
4	Koniec dla każdego
5	$d[v] \leftarrow 0$
6	$Q \leftarrow$ kolejka wierzchołków w grafie G
7	Dopóki $\boldsymbol{Q}$ nie jest puste
8	$\boldsymbol{u} \leftarrow$ zdejmij z kolejki Q wierzchołek o najmniejszej wartości $\boldsymbol{d[u]}$
9	Dla każdego wierzchołka $\boldsymbol{v}$, który jest sąsiadem u
10	Jeżeli $\boldsymbol{d[v] > \big(d[u] + w(u,v)\big)}$, gdzie $w(.)$ $\boldsymbol{oznacza}$ wagę połączenia $u \rightarrow v$
11	$\boldsymbol{d[v] \leftarrow d[u] + w(u,v)}$
12	$poprzednik[v] \leftarrow u$
13	Koniec jeżeli
14	Koniec dla każdego
15	Koniec dopóki
16	$\boldsymbol{ścieżka \leftarrow nowa\ lista}$
17	$\boldsymbol{ścieżka} \leftarrow$dodaj $\boldsymbol{s}$
18	$\boldsymbol{v \leftarrow d}$
19	Dopóki $\boldsymbol{poprzednik[v] \neq s}$
20	$\boldsymbol{v \leftarrow \text{poprzednik}[v]}$
21	$\boldsymbol{ścieżka} \leftarrow$dodaj $\boldsymbol{v}$
22	Koniec dopóki
23	$\boldsymbol{ścieżka} \leftarrow$dodaj $\boldsymbol{s}$

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Algorytm Dijkstra

1)



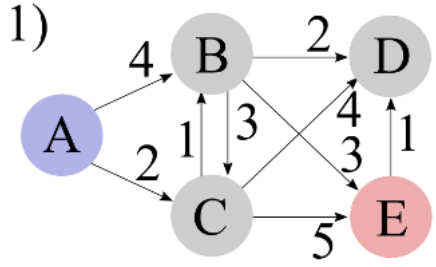
Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = ∞	Poprzednik[B] = -
d[C] = ∞	Poprzednik[C] = -
d[D] = ∞	Poprzednik[D] = -
d[E] = ∞	Poprzednik[E] = -

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

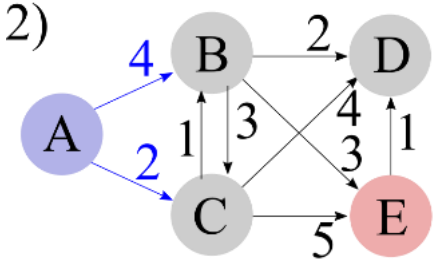
Algorytm Dijkstra

1)



Q = {A, B, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = ∞ Poprzednik[B] = -
d[C] = ∞ Poprzednik[C] = -
d[D] = ∞ Poprzednik[D] = -
d[E] = ∞ Poprzednik[E] = -

2)

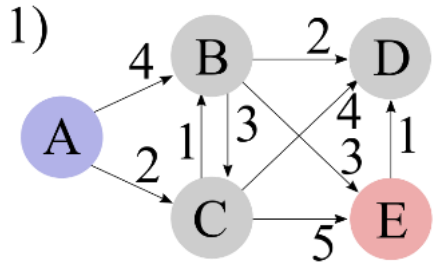


Q = {A, B, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = 4 Poprzednik[B] = A
d[C] = 2 Poprzednik[C] = A
d[D] = ∞ Poprzednik[D] = -
d[E] = ∞ Poprzednik[E] = -

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Algorytm Dijkstra

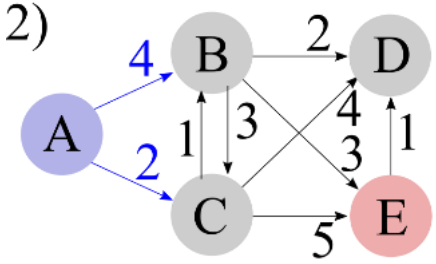
1)



Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = ∞	Poprzednik[B] = -
d[C] = ∞	Poprzednik[C] = -
d[D] = ∞	Poprzednik[D] = -
d[E] = ∞	Poprzednik[E] = -

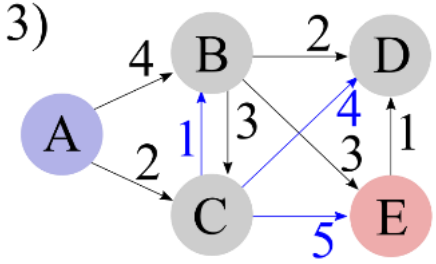
2)



Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = 4	Poprzednik[B] = A
d[C] = 2	Poprzednik[C] = A
d[D] = ∞	Poprzednik[D] = -
d[E] = ∞	Poprzednik[E] = -

3)



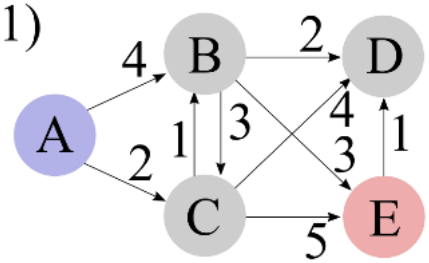
Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = 3	Poprzednik[B] = C
d[C] = 2	Poprzednik[C] = A
d[D] = 6	Poprzednik[D] = C
d[E] = 7	Poprzednik[E] = C

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

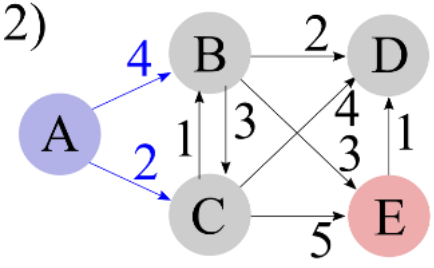
Algorytm Dijkstra

1)



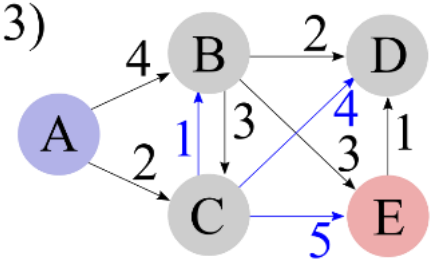
Q = {A, B, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = ∞ Poprzednik[B] = -
d[C] = ∞ Poprzednik[C] = -
d[D] = ∞ Poprzednik[D] = -
d[E] = ∞ Poprzednik[E] = -

2)



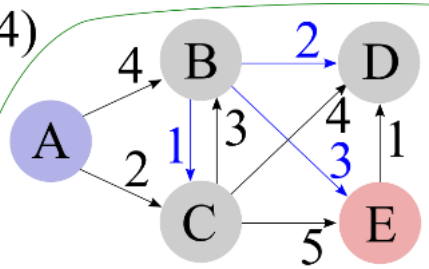
Q = {A, B, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = 4 Poprzednik[B] = A
d[C] = 2 Poprzednik[C] = A
d[D] = ∞ Poprzednik[D] = -
d[E] = ∞ Poprzednik[E] = -

3)



Q = {A, B, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = 3 Poprzednik[B] = C
d[C] = 2 Poprzednik[C] = A
d[D] = 6 Poprzednik[D] = C
d[E] = 7 Poprzednik[E] = C

4)

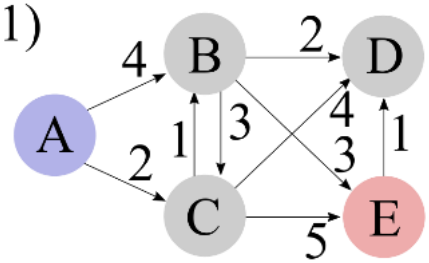


Q = {A, B, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = 4 Poprzednik[B] = C
d[C] = 2 Poprzednik[C] = A
d[D] = 6 Poprzednik[D] = C
d[E] = 7 Poprzednik[E] = C

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Algorytm Dijkstra

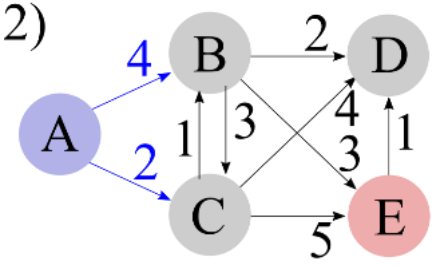
1)



Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = ∞	Poprzednik[B] = -
d[C] = ∞	Poprzednik[C] = -
d[D] = ∞	Poprzednik[D] = -
d[E] = ∞	Poprzednik[E] = -

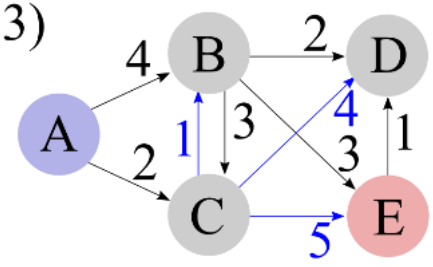
2)



Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = 4	Poprzednik[B] = A
d[C] = 2	Poprzednik[C] = A
d[D] = ∞	Poprzednik[D] = -
d[E] = ∞	Poprzednik[E] = -

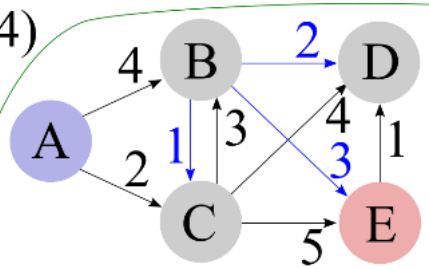
3)



Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = 3	Poprzednik[B] = C
d[C] = 2	Poprzednik[C] = A
d[D] = 6	Poprzednik[D] = C
d[E] = 7	Poprzednik[E] = C

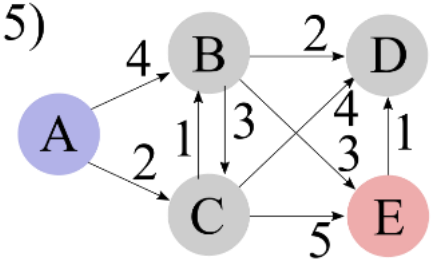
4)



Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = 4	Poprzednik[B] = C
d[C] = 2	Poprzednik[C] = A
d[D] = 6	Poprzednik[D] = C
d[E] = 7	Poprzednik[E] = C

5)



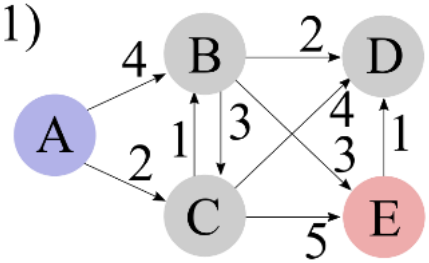
Q = {A, B, C, D, E}

d[A] = 0	Poprzednik[A] = -
d[B] = 4	Poprzednik[B] = C
d[C] = 2	Poprzednik[C] = A
d[D] = 6	Poprzednik[D] = C
d[E] = 7	Poprzednik[E] = C

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

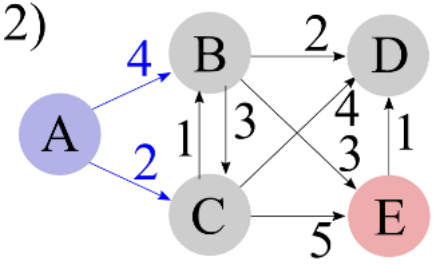
Algorytm Dijkstra

1)



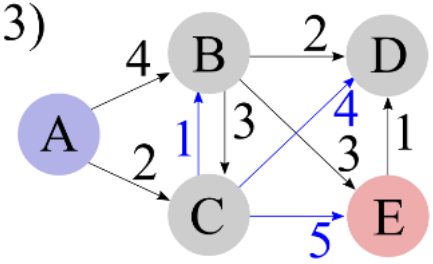
Q = {A, B, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = ∞ Poprzednik[B] = -
d[C] = ∞ Poprzednik[C] = -
d[D] = ∞ Poprzednik[D] = -
d[E] = ∞ Poprzednik[E] = -

2)



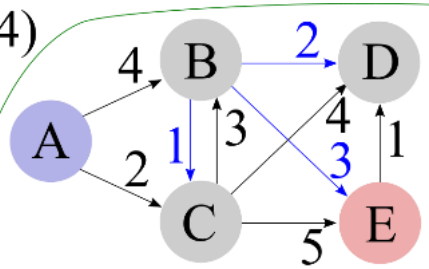
Q = {**A**, B, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = 4 Poprzednik[B] = A
d[C] = 2 Poprzednik[C] = A
d[D] = ∞ Poprzednik[D] = -
d[E] = ∞ Poprzednik[E] = -

3)



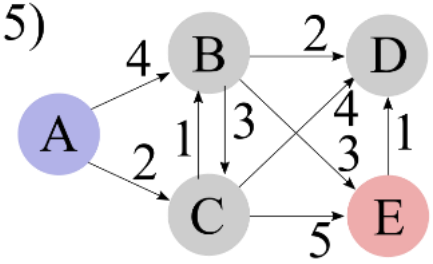
Q = {A, B, **C**, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = 3 Poprzednik[B] = C
d[C] = 2 Poprzednik[C] = A
d[D] = 6 Poprzednik[D] = C
d[E] = 7 Poprzednik[E] = C

4)



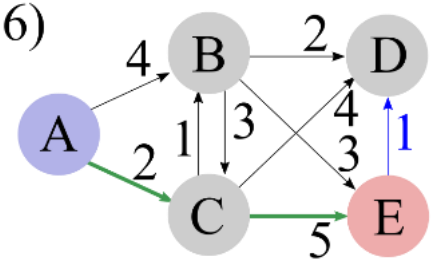
Q = {A, **B**, C, D, E}
d[A] = 0 Poprzednik[A] = -
d[B] = 4 Poprzednik[B] = C
d[C] = 2 Poprzednik[C] = A
d[D] = 6 Poprzednik[D] = C
d[E] = 7 Poprzednik[E] = C

5)



Q = {A, B, C, **D**, E}
d[A] = 0 Poprzednik[A] = -
d[B] = 4 Poprzednik[B] = C
d[C] = 2 Poprzednik[C] = A
d[D] = 6 Poprzednik[D] = C
d[E] = 7 Poprzednik[E] = C

6)

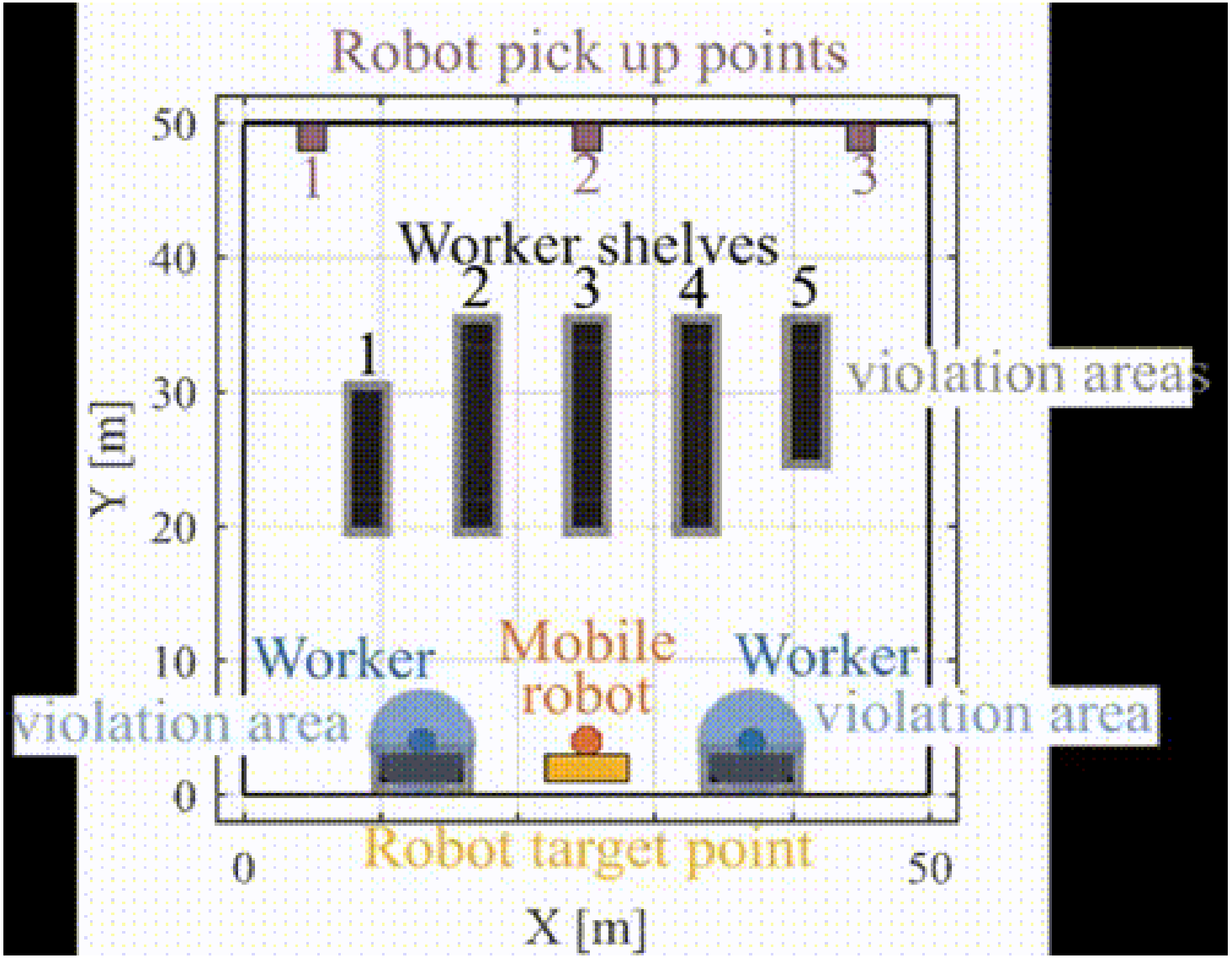


Q = {A, B, C, D, **E**}
d[A] = 0 Poprzednik[A] = -
d[B] = 4 Poprzednik[B] = C
d[C] = 2 Poprzednik[C] = **A**
d[D] = 6 Poprzednik[D] = C
d[E] = 7 Poprzednik[E] = **C**

Rozwiązanie: A-C-E

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Algorytm Dijkstra



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm A*

Jeżeli robot znajduje się na środku pokoju i ma dojechać w prawy górny róg to algorytm Dijkstra'y będzie sprawdzał również wierzchołki po przeciwległej stronie pomieszczenia, bo minimalizuje on odległość od wierzchołka startowego nie uwzględniając odległości do wierzchołka docelowego.

Nie poprawią one rozwiązania, ale algorytm jak było wspomniane bazuje na przeszukiwaniu drzewa wszerz, przez co algorytm analizuje wierzchołki idąc stosunkowo szeroką „falą".

Doprecyzowując algorytm Dijkstra'y do zagadnienia planowania ścieżki warto zaznaczyć, że wagi połączeń pomiędzy wierzchołkami grafu odpowiadają odległością pomiędzy punkami na mapie. Dzięki tej wiedzy możemy rozszerzyć analizowane wierzchołki o informację pomiędzy danym wierzchołkiem v, a wierzchołkiem docelowym d poprzez wyznaczenie odległości w linii prostej pomiędzy tymi wierzchołkami:

$$h(v) = \left|\left|d - v\right|\right|$$

Bardzo istotną cechę odległości pomiędzy tymi wierzchołkami jest to, że jest ona nie większa niż rzeczywista odległość możliwa do uzyskania w grafie (dla trzech punktów definiuje to nierówność trójkąta).

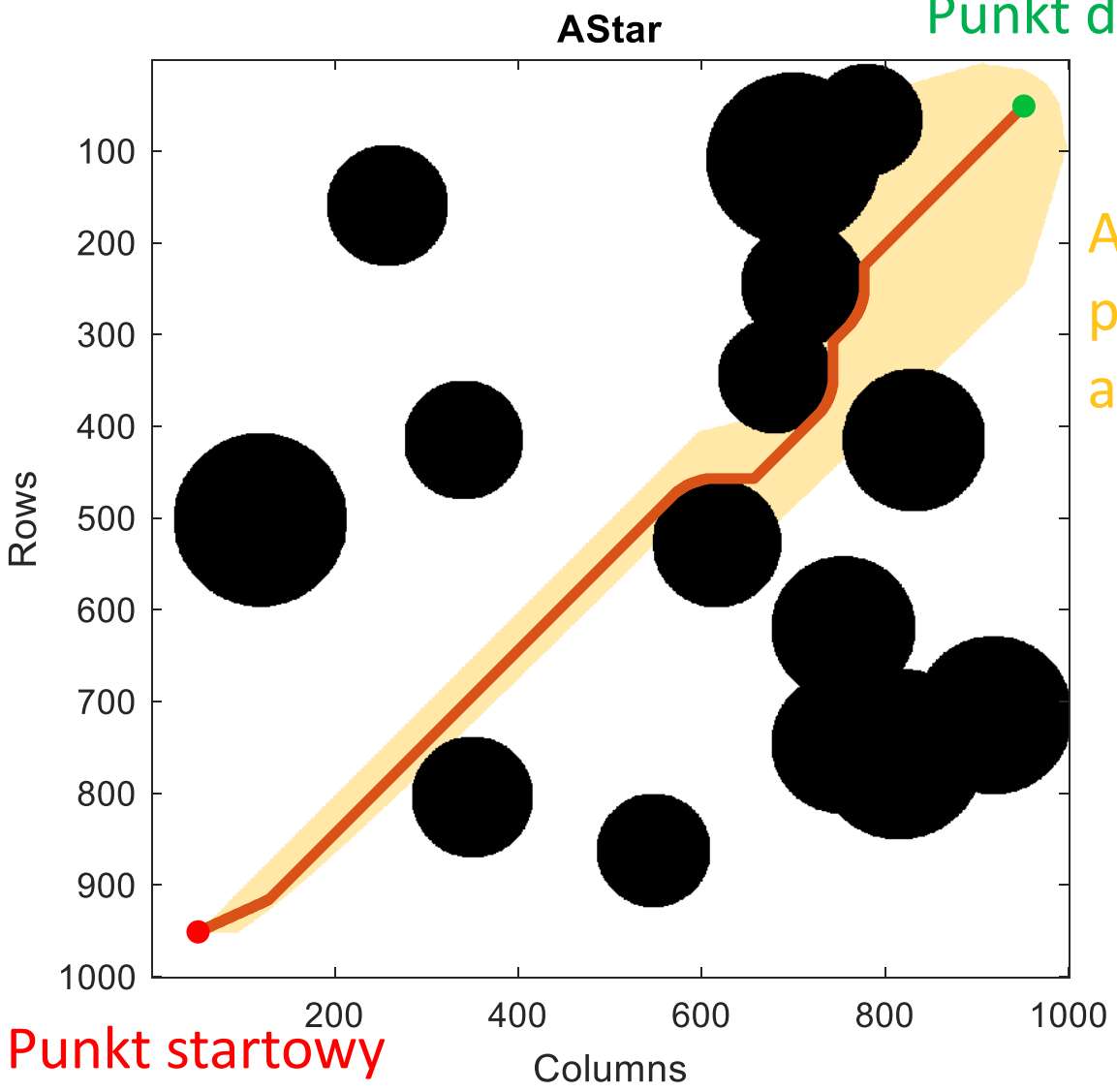
UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm A*

	Algorytm: Pseudokod algorytmu A*
	Wejście: G , s (wierzchołek początkowy), d (wierzchołek docelowy) Wyjście: ścieżka do celu
1	Dla każdego wierzchołka v w grafie G
2	$d[v] \leftarrow \infty$
3	$\text{poprzednik}[v] \leftarrow$ niezdefiniowano
4	Koniec dla każdego
5	$d[v] \leftarrow 0$
6	$Q \leftarrow$ kolejka wierzchołków w grafie G
7	Dopóki $\boldsymbol{Q}$ nie jest puste
8	$\boldsymbol{u} \leftarrow$ zdejmij z kolejki Q wierzchołek o najmniejszej wartości $\boldsymbol{d[u]}$
9	Dla każdego wierzchołka $\boldsymbol{v}$, który jest sąsiadem u
10	Jeżeli $\boldsymbol{d[v] > \left(d[u] + w(u,v) + h(v)\right)}$
11	$\boldsymbol{d[v] \leftarrow d[u] + w(u,v) + h(v)}$
12	$poprzednik[v] \leftarrow u$
13	Koniec jeżeli
14	Koniec dla każdego
15	Koniec dopóki
16	$\boldsymbol{ścieżka \leftarrow nowa\ lista}$
17	$\boldsymbol{ścieżka} \leftarrow$dodaj $\boldsymbol{s}$
18	$\boldsymbol{v \leftarrow d}$
19	Dopóki $\boldsymbol{poprzednik[v] \neq s}$
20	$\boldsymbol{v \leftarrow \text{poprzednik}[v]}$
21	$\boldsymbol{ścieżka} \leftarrow$dodaj $\boldsymbol{v}$
22	Koniec dopóki
23	$\boldsymbol{ścieżka} \leftarrow$dodaj $\boldsymbol{s}$

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Algorytm A*



Punkt docelowy

Analizowana pozycja przez algorytm A*

Punkt startowy

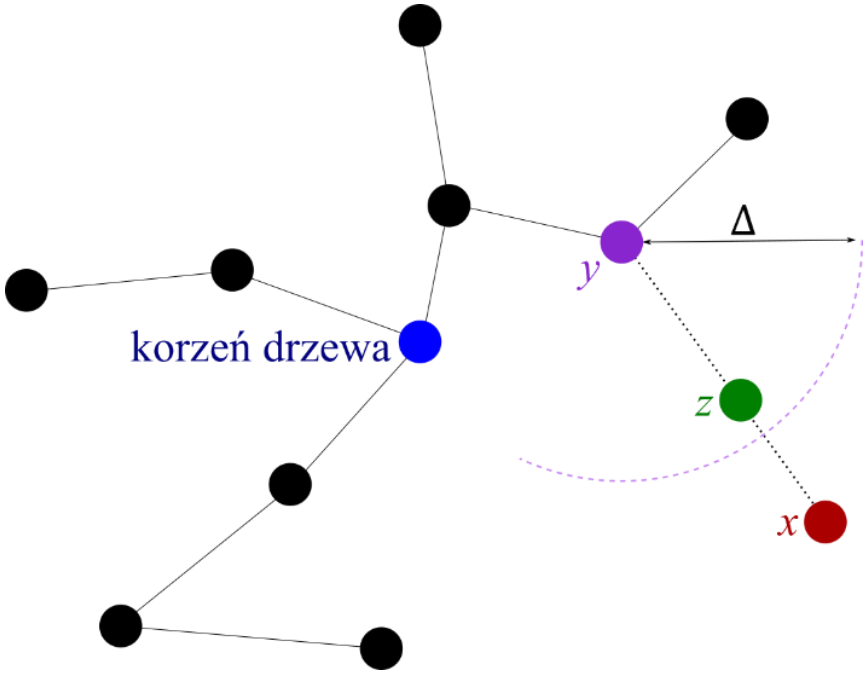
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Algorytm eksplorowania drzew losowych

Algorytm eksplorowania drzew losowych (RRT) polega na konstrukcji grafu w sposób losowy. Algorytm RRT tworzy specjalny rodzaj grafu zwany drzewem.

Jest to struktura, która posiada korzeń, który w naszym przypadku jest aktualną pozycją robota, i każdy kolejny wierzchołek w drzewie posiada dokładnie jednego rodzica.

Generując odpowiednio nowe punkty i łącząc je z drzewem, algorytm RRT dokonuje eksploracji całej mapy w celu znalezienia drogi do punktu docelowego.



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm eksplorowania drzew losowych

	Algorytm: Pseudokod budowy drzewa algorytmu eksplorowania drzew losowych
	Wejście: s (pozycja początkowa) Wyjście: T (drzewo eksplorujące mapę)
1	Stwórz nowe drzewo T i dodaj s jako jego korzeń
2	Dla $\boldsymbol{i = 1..N}$ wykonaj
3	Wygeneruj losową pozycję x
4	Jeżeli $\boldsymbol{CollisionCheck(x) = false}$
5	Znajdź wierzchołek y, który jest najbliższym sąsiadem w drzewie T dla x
6	Jeżeli $Distance(\boldsymbol{x}, \boldsymbol{y}) > \Delta$
7	Znajdź pozycję z, która leży na ścieżce łączącej x i y oraz spełnia zależność $Distance(z, y) \leq \Delta$
8	$\boldsymbol{x \leftarrow z}$
9	Koniec jeżeli
10	Jeżeli $LocalPlanner(x, y) = true$
11	Dodaj pozycję x do drzewa T ustawiając y jako jego rodzica
12	Koniec jeżeli
13	Koniec jeżeli
14	Koniec dla

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

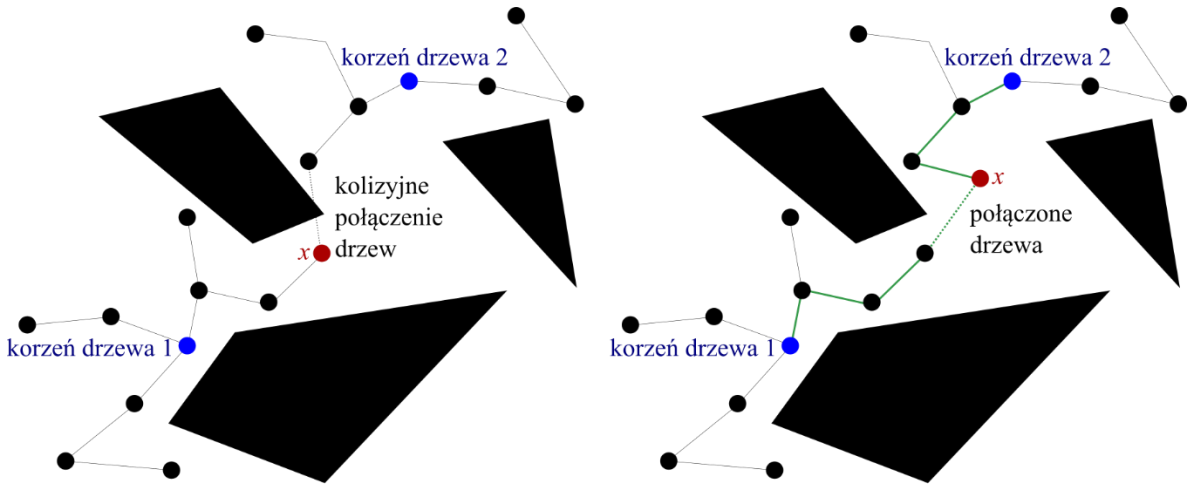
Algorytm eksplorowania drzew losowych

Źródło: https://en.wikipedia.org/wiki/Rapidly-exploring_random_tree

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm eksplorowania drzew losowych

	Algorytm: Pseudokod algorytmu eksplorowania drzew losowych
	Wejście: s (pozycja początkowa), d (pozycja docelowa) Wyjście: ścieżka do celu
1	Stwórz nowe drzewo A i dodaj s jako jego korzeń
2	Stwórz nowe drzewo B i dodaj d jako jego korzeń
3	Dopóki drzewa A i B nie są połączone
4	Rozszerz drzewo A o pozycję x
5	Znajdź najbliższego sąsiada y pozycji x w drzewie B
6	Jeżeli $LocalPlanner(x, y) = true$
7	Zwróć ścieżkę łączącą pozycje wierzchołków: - od punktu x do korzenia w drzewie A - od punktu y do korzenia w drzewie B
8	Koniec jeżeli
9	Zamień drzewa A i B
10	Koniec dopóki



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm mrówkowy

(ang. *Ant Colony Optimization algorithm*, ACO)

Jest to algorytm metaheurystyczny bazujący na populacji opracowany do zastosowań w problemach optymalizacyjnych. ACO jest inspirowany naturą, a dokładniej mrówkami.

Mrówka zostawia feromony podczas marszu. A wybór ścieżki, którą pójdzie mrówka zależy w pewnym stopniu od wartości feromonów na ścieżce. W związku z tym, mrówki początkowo w nowym terenie wybierają ścieżki bardziej losowo (brak feromonów) i z biegiem czasu poprzez zostawiany ślad feromonowy na ścieżkach zaczynają poruszać się po najlepszej ścieżce.

Pozostawianie feromonu na ścieżce łączącej wierzchołki grafu i i j przez k-tą mrówkę modelujemy jako odwrotnie proporcjonalne do długości ścieżki jaką mrówka pokonała aby dotrzeć z punktu początkowego do punktu końcowego:

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{Q}{L_k}, & jeżeli\ k-ta\ mrówka\ przeszła\ krawędzią\ \ i,j \\ 0, & w\ pozostałych\ przypadkach \end{cases}$$

gdzie: Q jest stałą (np. Q = 1), a Lk jest długością całej trasy od wierzchołka początkowego do końcowego mrówki.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm mrówkowy

(ang. *Ant Colony Optimization algorithm*, ACO)

Dodatkowo, feromon paruje, co jest symulowane poprzez zmniejszanie aktualnej wartości feromonu na ścieżce w każdej iteracji. Pełny model feromonu na danej ścieżce jest następujący:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \sum_{k=1}^{N} \Delta\tau_{i,j}^{k}$$

gdzie: ρ jest współczynnikiem parowania feromonu na ścieżce ($0 \leq \rho \leq 1$), N oznacza liczbę agentów (mrówek) algorytmu.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm mrówkowy

(ang. *Ant Colony Optimization algorithm*, ACO)

Kolejnym elementem kluczowym algorytmu jest decyzja o wyborze kolejnego wierzchołka. Kluczową rolę odgrywają wartość feromonu połączenia ($\tau_{i,j}$) oraz celowość zmiany stanu $\eta_{i,j}$, co w zakładanym przypadku planowania ścieżki można przyjąć za odwrotność odległości pomiędzy wierzchołkami $\left(\eta_{i,j} = \frac{1}{d_{i,j}}\right)$. Wzór na prawdopodobieństwo wyboru połączenia z wierzchołka i do wierzchołka j można zapisać następująco:

$$p_{i,j} = \frac{\tau_{i,j}^{\alpha} \cdot \eta_{i,j}^{\beta}}{\sum_{m \in dostępne wierzchołki} \tau_{i,m}^{\alpha} \cdot \eta_{i,m}^{\beta}}$$

gdzie: : α jest parametrem wpływu feromonu na ścieżce $\alpha \geq 0$), a β jest parametrem wpływu długości połączenia ($\beta \geq 1$)).

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm mrówkowy

(ang. *Ant Colony Optimization algorithm*, ACO)

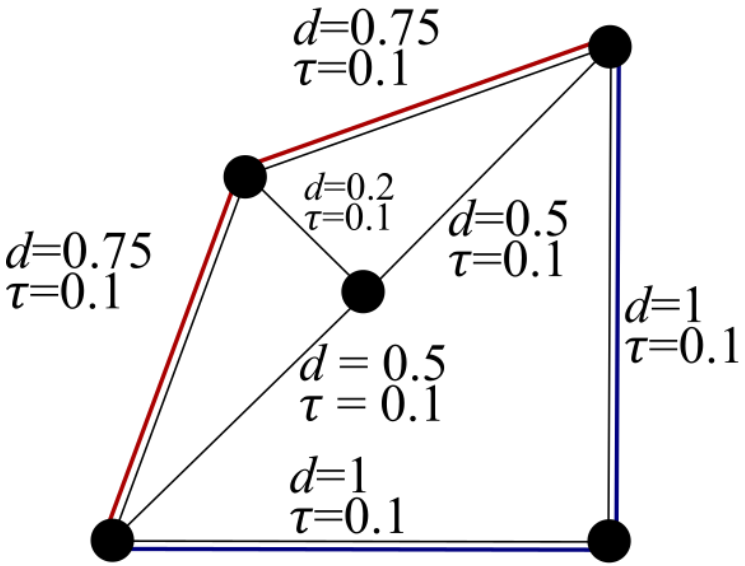
	Algorytm 7: Pseudokod algorytmu mrówkowego
	Wejście: s (pozycja początkowa), d (pozycja docelowa), N (populacja mrówek), M (liczba iteracji), Q, α, β, ρ (parametry) Wyjście: ścieżka do celu
1	Inicjalizacja zmiennych
2	Dla $iter = 1 \ldots M$ wykonaj
3	Dla $\boldsymbol{k = 1..N}$ wykonaj
4	Pozycja mrówki $\leftarrow s$
5	Dopóki mrówka nie osiągnęła punktu docelowego (d)
6	Oblicz prawdopodobieństwo dla każdego możliwego ruchu
7	Wybierz kolejną pozycję zgodnie z prawdopodobieństwem
8	Koniec dopóki
9	Aktualizuj ilość feromonów na ścieżkach
10	Koniec dla
11	Koniec dla
12	Zwróć ścieżkę wybierając trasę o największych prawdopodobieństwach

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Algorytm mrówkowy

(ang. *Ant Colony Optimization algorithm*, ACO)

Trasy mrówek



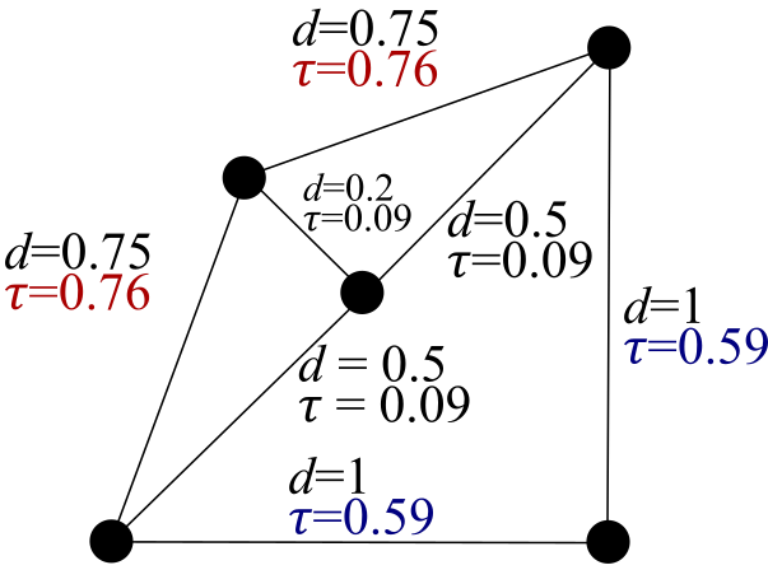
$L_1 = 0.75 + 0.75 = 1.5$

$\Delta\tau^1 = 1/1.5 = 0.67$

$L_2 = 1 + 1 = 2$

$\Delta\tau^2 = 1/2 = 0.5$

Aktualizacja feromonu



$\rho = 0.1$

$(1-0.1) * 0.1 + 0 = 0.09$

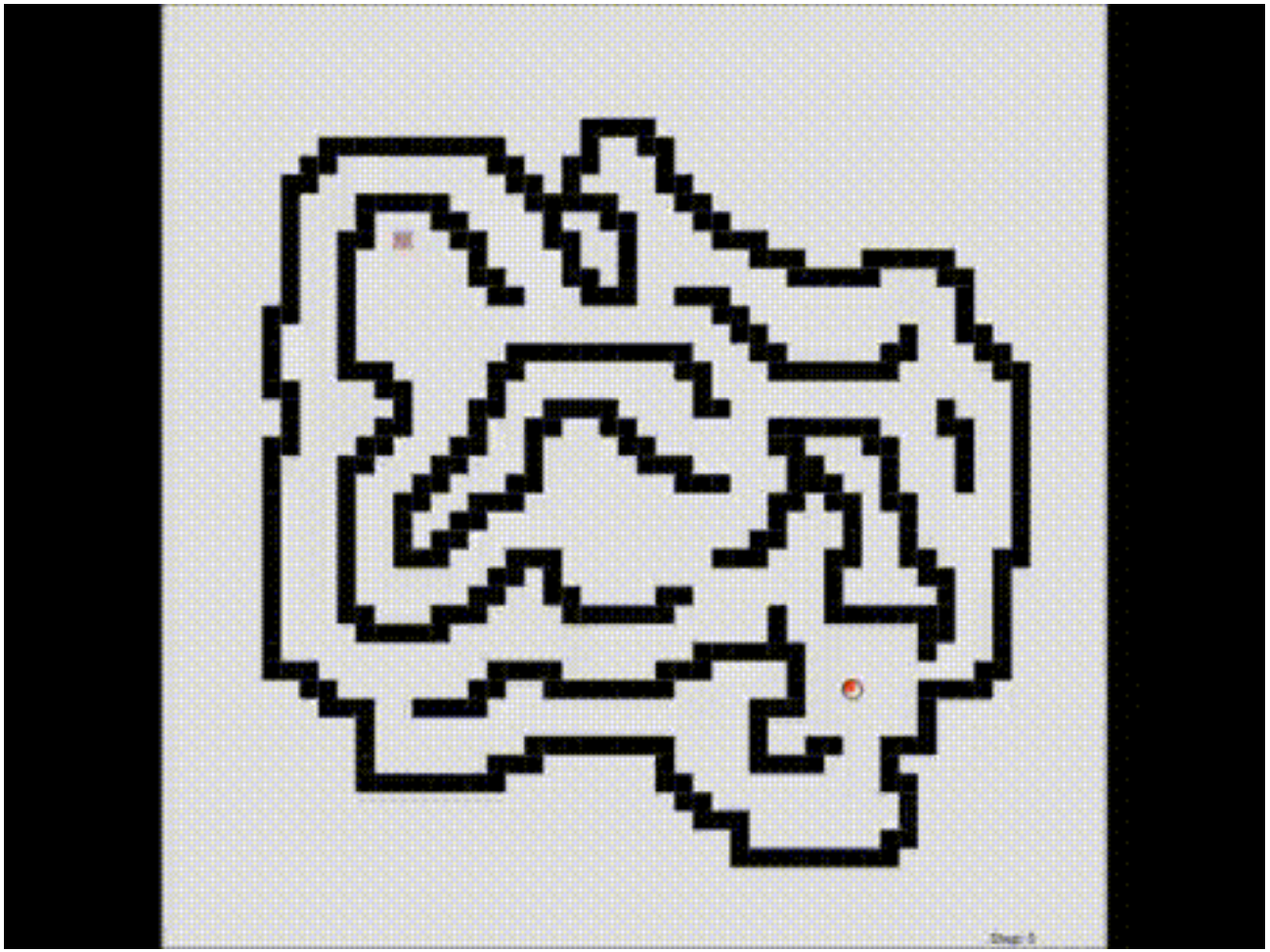
$(1-0.1) * 0.1 + 0.67 = 0.76$

$(1-0.1) * 0.1 + 0.5 = 0.59$

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Algorytm mrówkowy

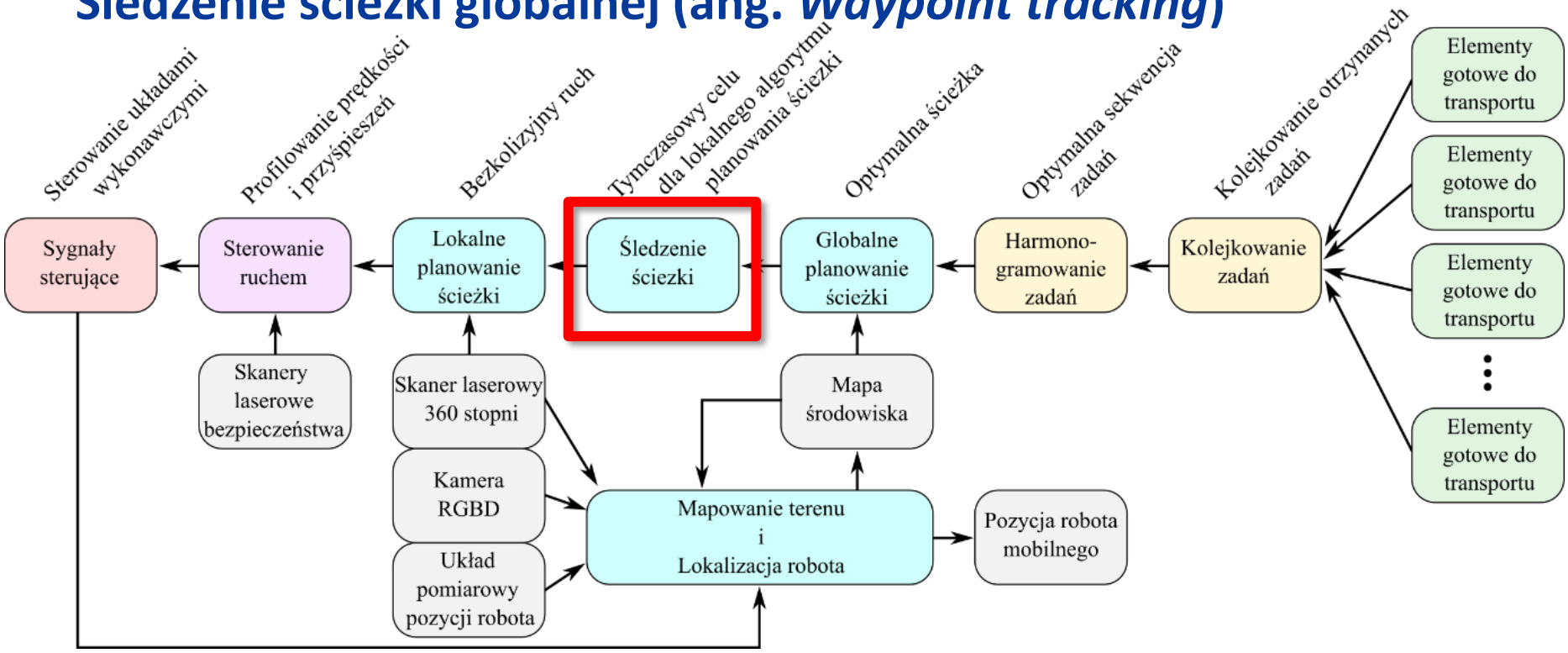
(ang. *Ant Colony Optimization algorithm*, ACO)



Źródło: https://www.youtube.com/watch?v=SJM3er3L6P4

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)



Założenia:

- Znajomość pozycji robota
- Znajomość ścieżki
- Znajomość mapy środowiska
- Dane z lokalnych czujników robota

Oczekiwania:

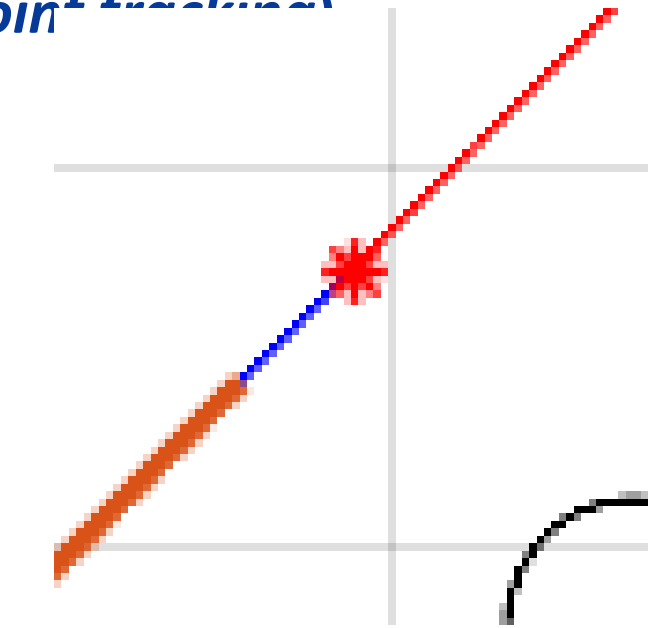
- Śledzenie globalnej ścieżki

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)

Algorytmy śledzenia ścieżki globalnej tj. Pure Pursuit nie posiadają wbudowanych mechanizmów omijania przeszkód wykorzystując czujniki pokładowe.

W celu bezkolizyjnej realizacji globalnej ścieżki należy zastosowań algorytm planowania ścieżki lokalnej.



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

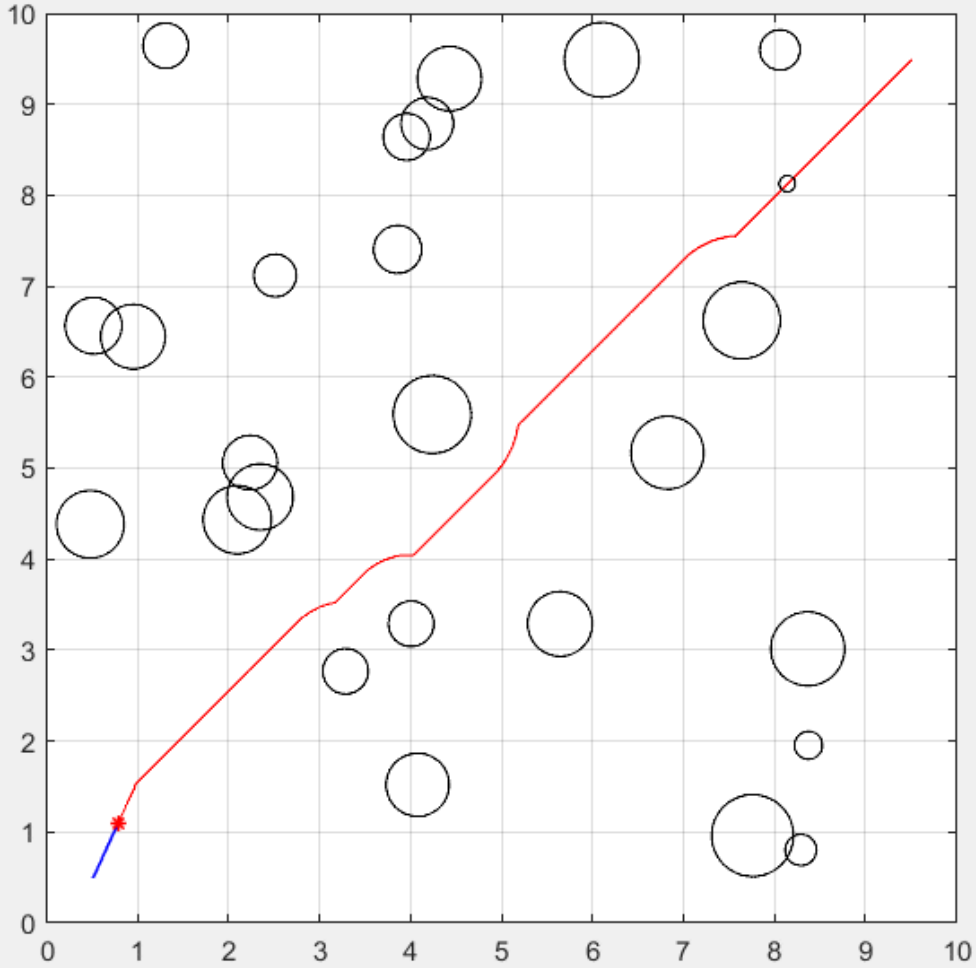
Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)

Wyprzedzenie o np. 1m ścieżki jako punkt do śledzenia dla algorytmu planowania ścieżki lokalnej

Najbliższy punkt ścieżki globalnej od aktualnej pozycji robota

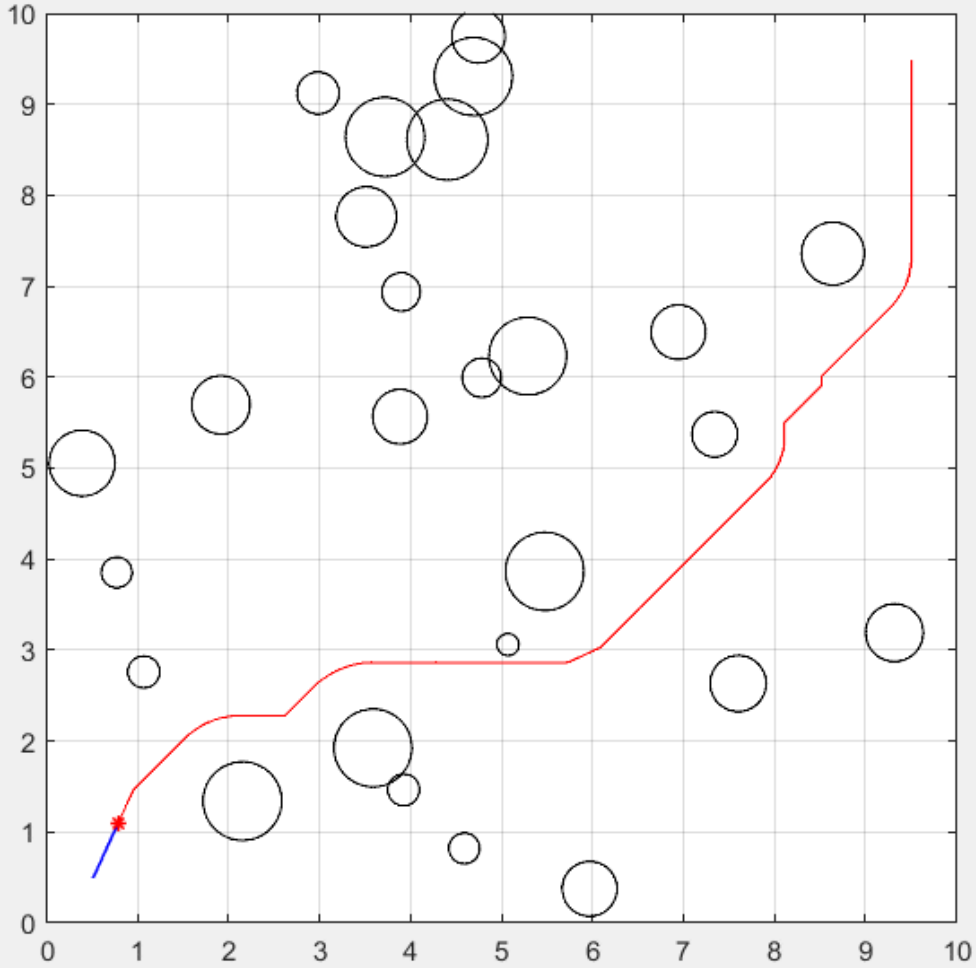
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Śledzenie ścieżki globalnej (ang. *Waypoint tracking*) + APF



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Śledzenie ścieżki globalnej (ang. *Waypoint tracking*) + APF



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Śledzenie ścieżki globalnej (ang. *Waypoint tracking*) + APF