

Roboty mobilne

1. Planowanie ścieżki - c.d.

2. Sterowanie

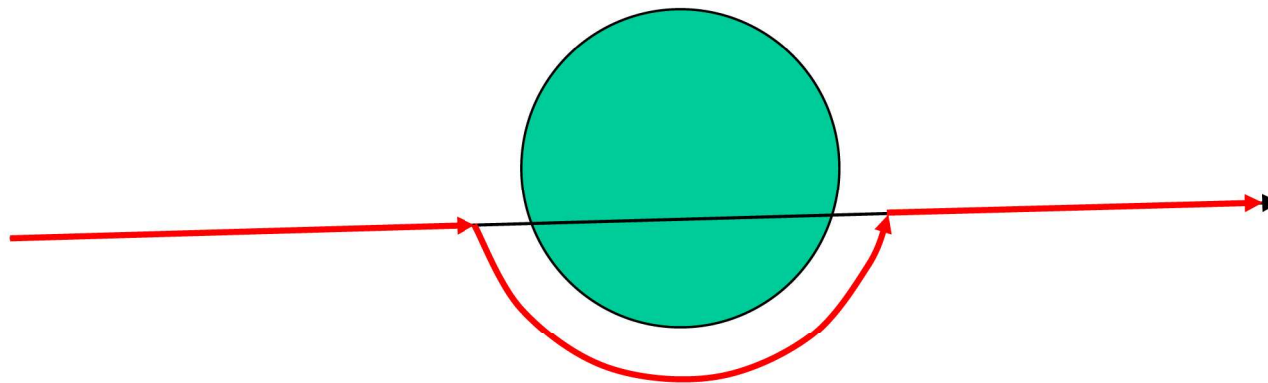
Metody planowania lokalnego

- Cel metod: unikanie kolizji z przeszkodami
- Są realizowane jako niezależne podzadania w systemach sterowania ruchem robotów
- Służą do wspomagania metod globalnych: zakłada się istnienie nadrzędnej metody planowania ścieżki do celu
- Są przydatne wtedy, gdy planowanie zostało wykonane na mapie wstępnej, bez uwzględniania aktualnych odczytów z sensorów

Planowanie lokalne

Algorytm *robaka*

- Po dojściu do przeszkody iść wzdłuż jej brzegu tak długo, aż dotrzemy do:
 - powtórnego spotkania poprzedniego planu
 - znalezionej nowej ścieżki do celu
- Odpowiednie algorytmy wyboru kierunku ruchu



Planowanie lokalne

- Obserwacja pola 360 st. wokół robota
- Dodanie funkcji (argumentem jest kąt):
 - zgodności z kierunkiem do celu
 - odległości od przeszkód
 - poprzednich parametrów ruchu (aktualny kierunek ruchu)
- Wybór kierunku najlepszego

Planowanie lokalne

Inne:

- Dodanie więzów do przestrzeni prędkości (v, ω)
- Dodatkowe powiększenie przeszkód przy planowaniu globalnym
- Dokładne modele matematyczne

Inne metody planowania ruchu

Algorytmy genetyczne

Wzorowane na ewolucji organizmów żywych

Początek:

- populacja rozwiązań (zwykle przypadkowych)
- funkcja przystosowania (pozwala ocenić, które rozwiązanie jest lepsze od pozostałych)

Poszukujemy takiego rozwiązania, które będzie miało najwyższą wartość funkcji przystosowania - będzie miało najwyższą jakość

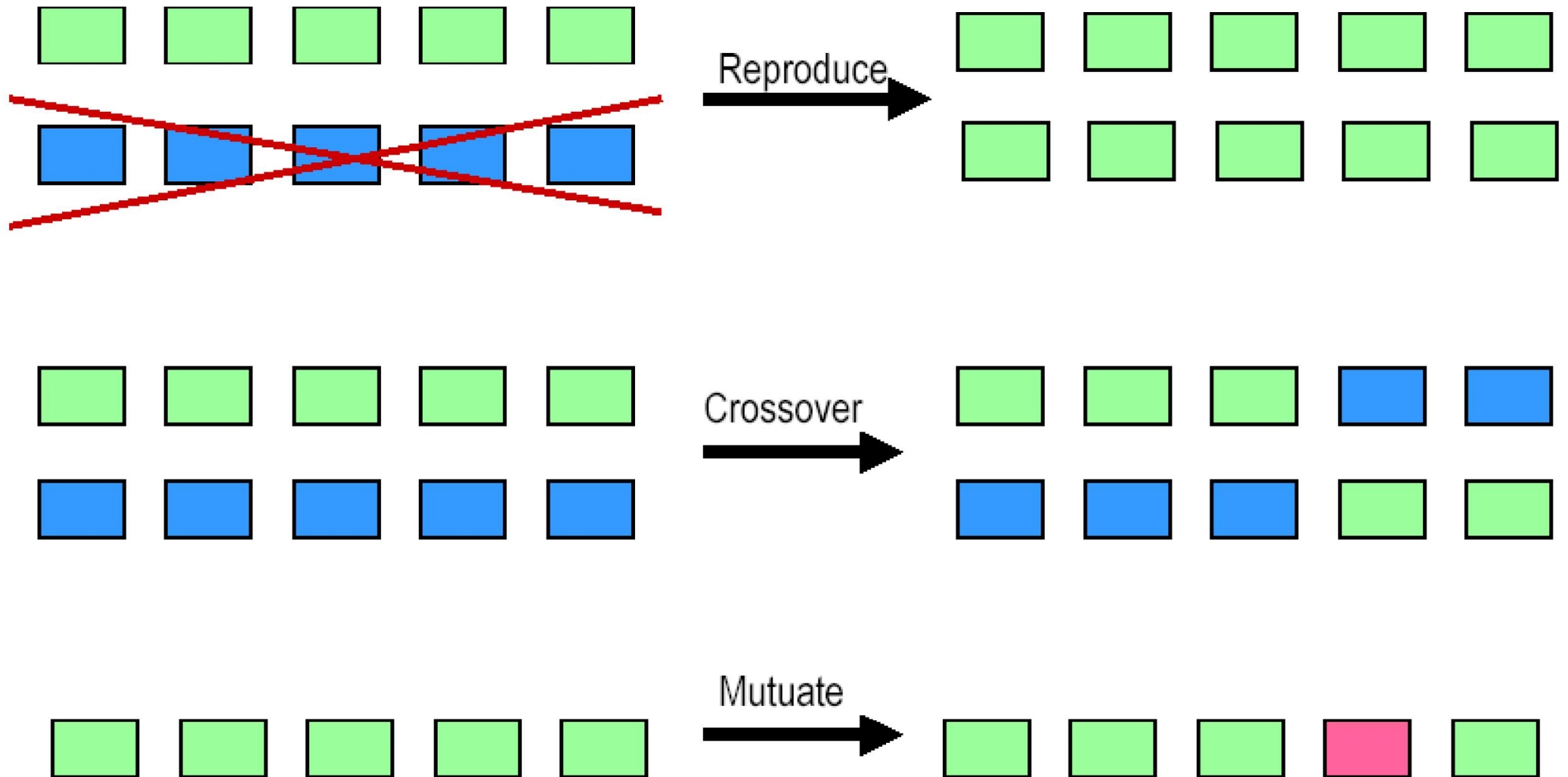
Problem: sformalizowanie rozwiązań

Algorytm genetyczny

Proces otrzymywania następnego pokolenia:

- wykonanie na populacji operacji:
 - rozmnożenia
 - krzyżowania
 - mutacji
- powyższe działania wykonuje się *losowo*
- odrzucenie z populacji *osobników* najgorzej spełniających swoje zadania (najsłabsze wyniki funkcji przystosowania) i pozostawienie do dalszych działań tych najlepszych

Operatory genetyczne



Algorytmy genetyczne

- Potrzeba bardzo wielu iteracji
- Obliczenia trwają bardzo długo
- Nie zawsze udaje się uzyskać żądane wyniki
- Element losowości algorytmu skutkuje brakiem deterministycznych wyników (nie da się np. udowodnić poprawności algorytmu)
- Bardzo ważne (i trudne) poprawne sformułowanie funkcji dopasowania populacji
- Nie ma jeszcze aplikacji w rzeczywistych robotach

Przykład planowania genetycznego

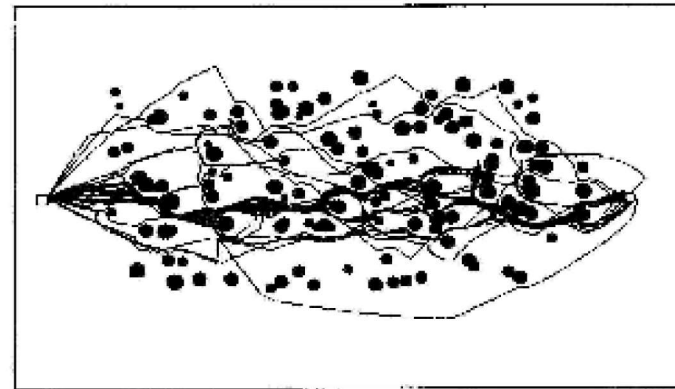
Poszukiwanie ścieżki:

a) najbezpieczniejszej

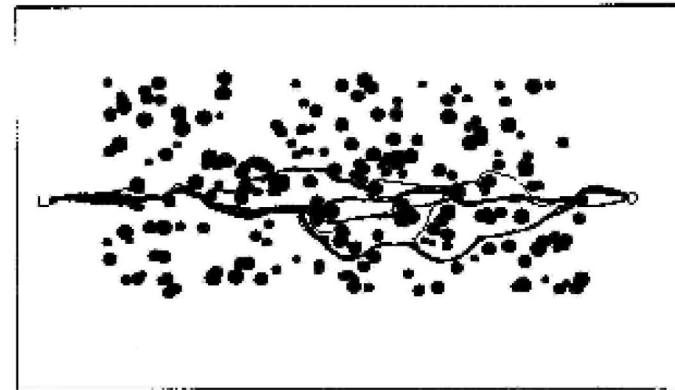
b) najszybszej

c) najprostszej

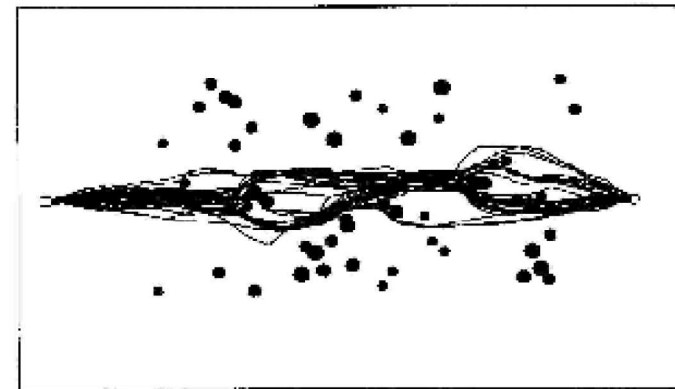
Na rysunkach są pokazane
przykładowe populacje
przedstawiające ścieżki



(A)



(B)



(C)

Algorytmy mrówkowe

- Cel kolonii mrówek: przeżycie całej kolonii
- działanie pojedynczej mrówki - proste
- „działanie” kolonii - dość skomplikowane
- widoczna specjalizacja osobników
- priorytet: dobro stada

Algorytmy mrówkowe

- Mrówka idąc pozostawia za sobą ślad zapachowy - feromonowy (związek chemiczny wyczuwalny przez owady)
- Feromon po jakimś czasie przestaje być wyczuwalny (wietrzeje)
- Mrówka po obszarze bez śladów innych mrówek porusza się losowo
- Jeżeli znajdzie ślad, to stara się iść po tym śladzie
- Ślad jest łatwiej wykrywalny przez mrówki wtedy, gdy zostawi go większa liczba owadów
- Mrówka mimo śladu niekiedy porusza się losowo

Algorytmy mrówkowe

- Wspomaganie nawigacji: znaczniki (słońce i ziemskie pole magnetyczne), „odometria” (mrówka z grubsza ocenia odległość, którą przeszła)
- Cel wędrówki mrówki: znalezienie jedzenia i zanieśenie go do mrowiska
 - znalezienie nowego źródła pożywienia
 - znalezienie tego źródła, z którego czerpią inne
- Mrówka po znalezieniu poszukiwanego obiektu stara się wrócić do mrowiska, wtedy zostawia najmocniejszy ślad feromonowy

Algorytmy mrówkowe

Przykład:

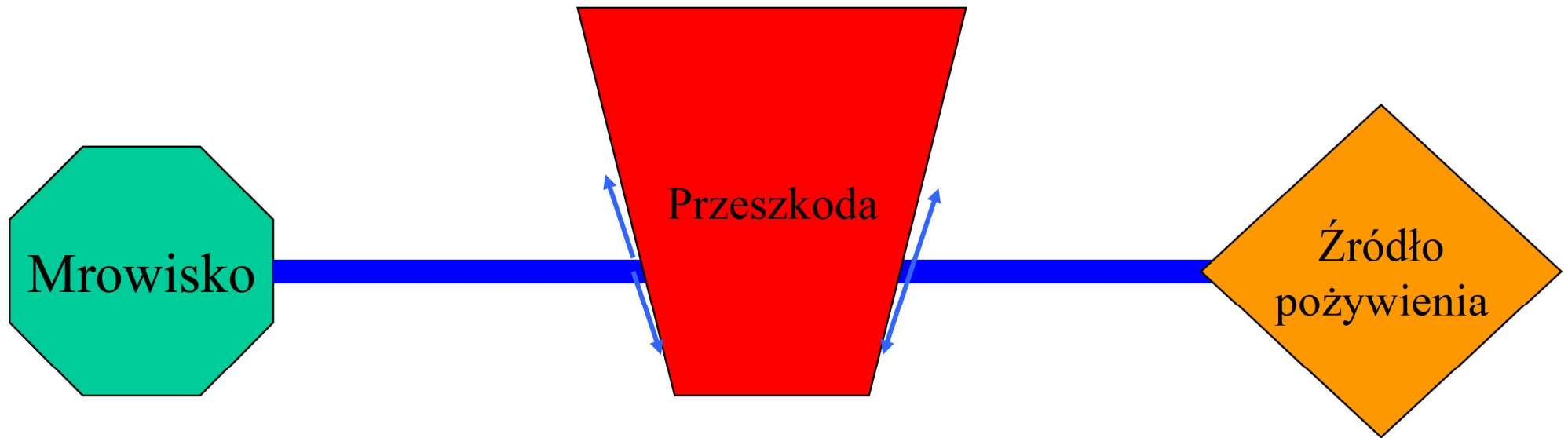
- z mrowiska rusza na poszukiwania wiele mrówek
- te, które znajdują coś ciekawego, starają się wrócić do mrowiska
- następne idą korzystając ze słabych śladów
- najwięcej mrówek wraca z tych miejsc, gdzie jedzenie jest najbliżej, one też najbardziej wzmacniają ślad
- tworzy się kilka najkrótszych dróg, łączących mrowisko z najbliższymi źródłami pożywienia

Przykład omijania przeszkody



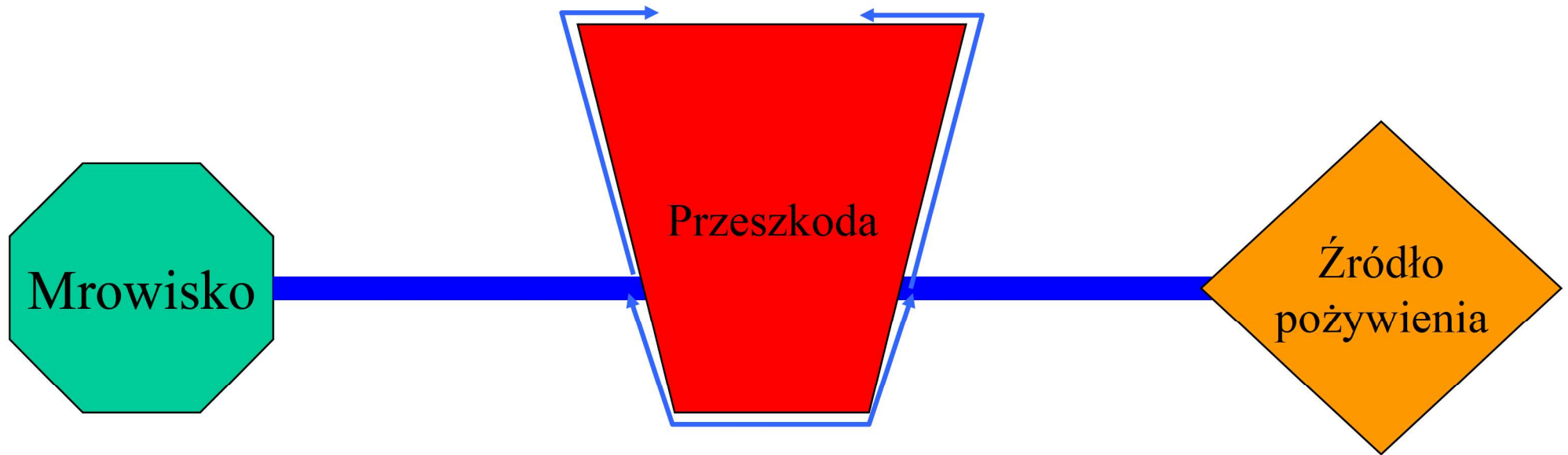
- z mrowiska do celu jest znaleziona najkrótsza ścieżka (ślad feromonowy), po której w obu kierunkach poruszają się mrówki

Omijanie przeszkody



- na ścieżce pojawia się przeszkoda (ginie ślad feromonowy)
- mrówki starają się obejść przeszkodę (wybór kierunku jest losowy)

Omijanie przeszkody



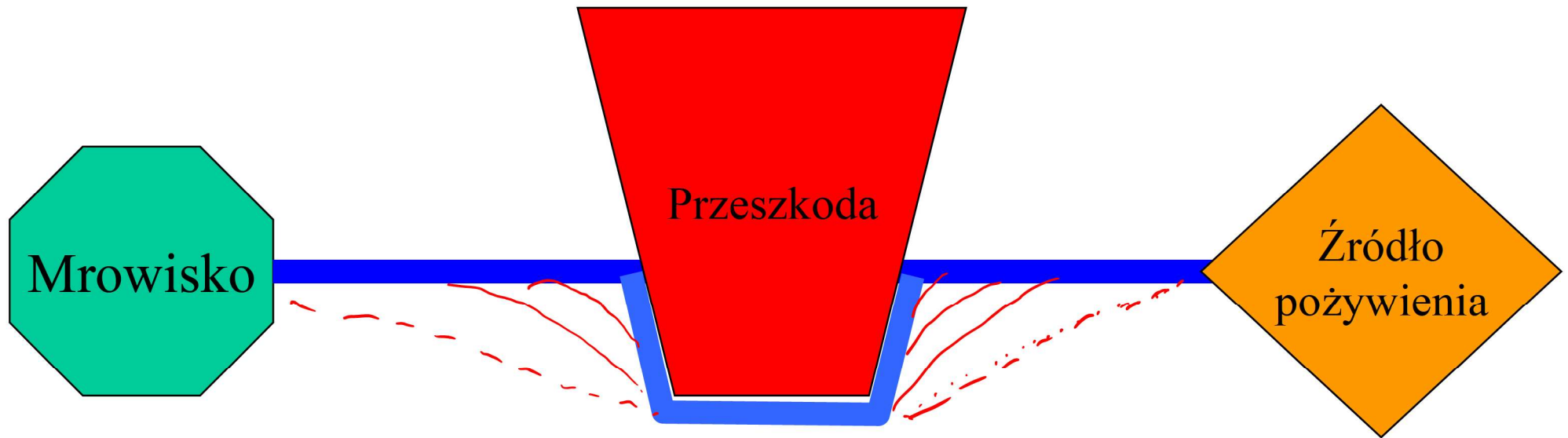
- na właściwą ścieżkę wracają szybciej te mrówki, które wybrały krótszą drogę obejścia przeszkody

Omijanie przeszkody



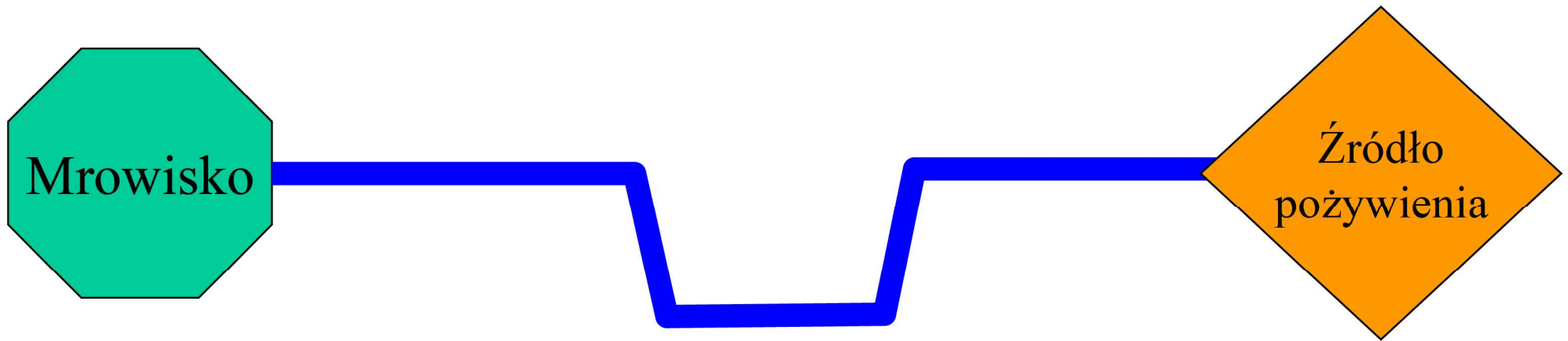
- po przejściu większej liczby mrówek ślad na ścieżce krótszej jest mocniejszy, bo pierwsze mrówki wcześniej wróciły - statystycznie liczba mrówek na jednostkę długości była większa

Omijanie przeszkody



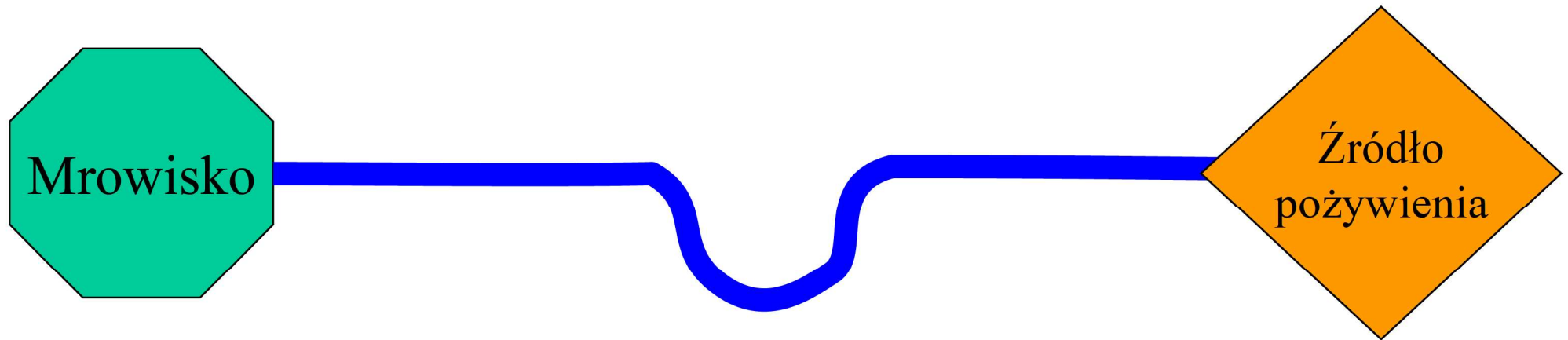
- ostatecznie wszystkie mrówki poruszają się po nowej (najkrótszej) ścieżce, ślad feromonowy na ścieżce dłuższej zanika

Omijanie przeszkody



- po usunięciu przeszkody mrówki poruszają się jakiś czas starą ścieżką

Omijanie przeszkody



- losowe zaburzenia w ruchu poszczególnych owadów pozwalają po jakimś czasie „wyprostować” ścieżkę

Omijanie przeszkody



- w końcu mrówki znajdują nową najkrótszą ścieżkę

Algorytmy mrówkowe a robotyka

- Mrowisko - miejsce, gdzie jest robot
- Jedzenie - cel robota
- **Symulacyjnie** wypuszczamy kilka robotów, aby znalazły cel
- Po znalezieniu próbują wrócić
- Następne wirtualne roboty korzystają z pozostawionych przez poprzednie śladów i wzmacniają je
- Efekt: powinien być zbliżony do efektów mrówek - najkrótsza ścieżka do celu

Algorytmy mrówkowe w robotyce

Także jako uzupełnienie algorytmu planowania:

- pojawia się nieprzewidywalna przeszkoda
- symulacyjnie (lub nie - w przypadku zespołu robotów mobilnych) prowadzimy robota wg algorytmu mrówkowego
- po wielu iteracjach znajdujemy najkrótszą ścieżkę

Ciekawe aspekty algorytmów mrówkowych

- Pojedyncza mrówka może dojść do celu
- Zespół wykazuje tendencje do optymalizacji
- Rozproszony system optymalizacyjny z małym wkładem każdego z osobników

W algorytmach ustala się:

- modele „wyparowywania” feromonu
- modele wyboru kierunku
- modele zaburzeń losowych

Algorytmy są bardzo czasochłonne

Logika rozmyta

Problem: Jak sklasyfikować np. temperaturę w mieszkaniu:

<i>bardzo zimno</i>	-5 ... 10
<i>zimno</i>	5 ... 14
<i>chłodno</i>	12 ... 20
<i>optymalnie</i>	16 ... 25
<i>ciepło</i>	18 ... 30
<i>gorąco</i>	24 ... 35
<i>bardzo gorąco</i>	30 ...

Logika rozmyta

- Dostarcza narzędzi do prawidłowego ustalania reguł w warunkach wiedzy rozmytej, np.:
 - *kiedy jest chłodno **lub** zimno **lub** bardzo zimno, włącz ogrzewanie*
- Narzędzia: funkcje logiki rozmytej
- Funkcja *przynależności*: stopień podobieństwa wartości wejściowej do zbiorów rozmytych
- Możliwości: dobre działanie w warunkach wiedzy niepełnej lub zakłóconej - czyli tego, z czym musi uporać się programista robotów mobilnych

Logika rozmyta w robotyce

- Wczesne rozwiązania - sterowanie na zasadzie reguł:
 - nie działało prawidłowo przy zakłóceniach
- Możliwość określenia niedokładnych (*rozmytych*) reguł działania robota
- Robot może mieć np. kilka celów i za pomocą sterowania rozmytego podejmować decyzje, kiedy do którego z nich się udać