

www.kwmimkm.polsl.pl

OBLICZENIA EWOLUCYJNE

wykład 6

Start

Losowe generowanie populacji początkowej

Generowanie populacji potomnej

Selekcja Kąpielowa

Mutacja

SENDING CHROMOSOMES TO COMPUTERS AND RECEIVING FITNESS VALUE

TERMINATION CONDITION

YES

NO

END

subGenerations

www.kwmimkm.polsl.pl

OPTIMALIZACJA WIELOMODALNA

www.kwmimkm.polsl.pl

Cel:
Znalezienie pewnej liczby k jak najlepszych ekstremów lokalnych funkcji wielomodalnych.

Dlaczego?
Niekiedy dogodnie jest znalezienie kilku różnych rozwiązań, by móc wybrać takie, które jest najlepsze z punktu widzenia kryteriów nie zawartych w funkcji celu (np. estetyka rozwiązania).

3

www.kwmimkm.polsl.pl

Podstawowa technika: utrzymywanie różnorodności populacji bazowej.

Przeszkoda: globalność selekcji (każdy osobnik konkuruje podczas reprodukcji i sukcesji z każdym innym).

Zapobieganie:

- Dodanie czynnika losowego, niezależnego od stanu populacji (zwykle dodatkowy operator typu mutacji bądź losowe zaburzenie funkcji przystosowania),
- Zmniejszenie zasięgu selekcji (ograniczenie konkurencji do osobników znajdujących się blisko siebie w przestrzeni genotypów – wyłanianie nisz).

4

www.kwmimkm.polsl.pl

Techniki wyłaniania nisz:

- Metody polegające na wyróżnianiu podpopulacji na podstawie odległości genotypów;
- Metody koewolucyjne (równoczesna ewolucja wielu autonomicznych populacji z częściową wymianą mat. genet.);
- Metody dokonujące lokalnej deformacji funkcji przystosowania (np. dodawanie losowego szumu).

www.kwmimkm.polsl.pl

RÓWNOLEGŁOŚĆ W AE

6

1. Równoległość na poziomie implementacji:

(schemat algorytmu nie zmienia się):

- Równoległe generowanie osobników w populacji początkowej;
- Równoległe obliczanie funkcji przystosowania;
- Równoległe wykonywanie operatorów ewolucyjnych;
- Sekwencyjne dokonywanie wyboru nowej populacji (selekcja).



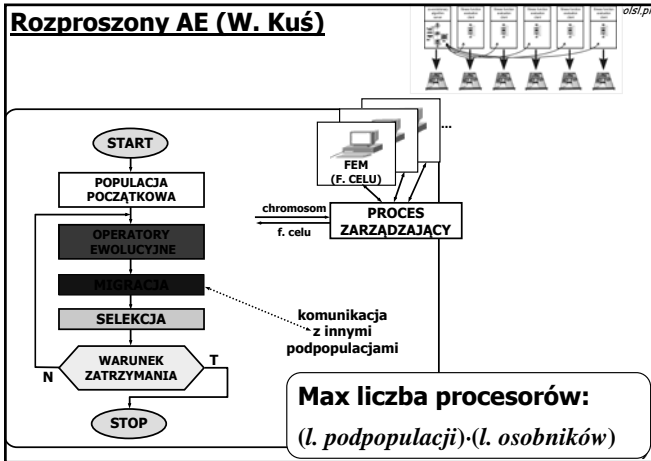
2. Równoległość na poziomie koncepcji

(algorytmy koewolucyjne):

- **Algorytm wyspowy**
 - Podpopulacje ewoluują (prawie) niezależnie, ze sporadyczną wymianą informacji.
 - Parametry ewolucji a nawet funkcja przystosowania mogą być różne w różnych podpopulacjach.
- **Algorytm komórkowy** (masowo równoległy, dyfuzyjny)
 - Każdy procesor zajmuje się ewolucją jednego osobnika, dla którego wyznaczane jest pewne sąsiedztwo w przestrzeni genotypów.
 - Ewolucja odbywa się z uwzględnieniem sąsiedztwa (krzyżowanie z losowymi sąsiadami).



Rozproszony AE (W. Kuś)



POŁĄCZENIE AE Z METODAMI LOKALNYMI



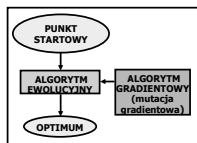
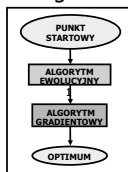
IDEA:

Połączenie nieprecyzyjnego AE z nieodpornymi metodami lokalnymi.

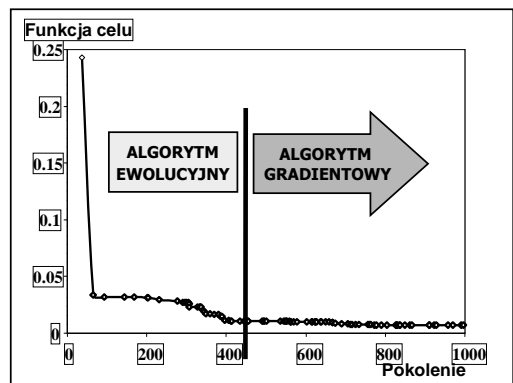
Prowadzi to często do uzyskania algorytmu hybrydowego o właściwościach lepszych od każdej z wchodzących w jego skład metod.

Warianty:

- algorytmy, w których metoda lokalna służy do dokończenia obliczeń rozpoczętych przez AE;
- wprowadzeniu dodatkowego operatora genetycznego (np. mutacji gradientowej), który sprowadzałby się do wykonania metody lokalnej;
- przeszukiwanie lokalne wykonywane podczas obliczania przystosowania osobnika.



Np. metody stosowane kolejno:

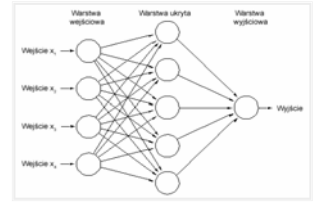


AE + Sztuczne sieci neuronowe



SZTUCZNE SIECI NEURONOWE:

- Nie wymagają programowania (tylko uczenie);
- Mają zdolność uogólniania;
- Są wysoce odporne na szумы i zniekształcenia sygnału;
- Pomagają wykrywać istotne powiązania pomiędzy danymi.



Stosuje się je gdy istnieje duża złożoność zagadnienia i trudno jest jednoznacznie określić formalne kryteria, dla stworzenia programu komputerowego.

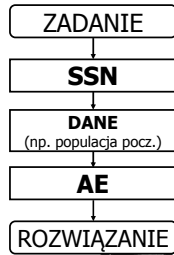


POŁĄCZENIE AE I SSN:

1. **Wspomagające** (metody stosowane kolejno);
2. **Współdziałające** (metody stosowane jednocześnie).

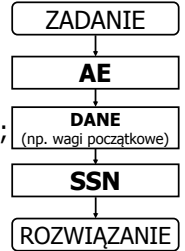
Ad. 1. (Połączenia wspomagające)
SSN do wspomagania AE (rzadziej).

- Zastosowanie: *system hybrydowy do rozwiązania problemu połączeń drogowych*:
SSN – utworzenie populacji pocz.
AE – procedura optymalizacyjna



AE do wspomagania SSN (część).

- a) AE do przygotowania danych dla SSN;
- b) AE do wyboru reguły uczenia lub parametrów sterujących uczeniem SSN;
- c) AE do analizy SSN – budowa narzędzi do wyjaśnienia działania SSN.



Ad. 2. (Połączenia współdziałające)

AE do uczenia SSN

- Optymalizacja wag w sieci o ustalonej topologii (*przy problemach z liczeniem pochodnych*).

AE do określania topologii SSN

- Optymalizacja architektury SSN – poszukiwanie architektury, która działa najlepiej dla danego zadania przy zadanym kryterium optymalności.

Systemy łączące adaptacyjne strategie SSN i AE

1. SSN do zadań optymalizacji i jednocześnie AE do ustawiania wag sieci
2. SSN realizuje operacje genetyczne.



INNE AE



PROGRAMOWANIE GENETYCZNE



19

programy, które powstają samoczynnie...

- Automatyczne generowanie tekstów programów, jeśli znane są kryteria oceny prawidłowości działania.
- Język bazowy – LISP (*program jest reprezentowany w identyczny sposób jak dane - w postaci drzewa*).
- Kodowanie binarne zastąpiono drzewiastym.
- W węzłach mogą znajdować się:
 - symbole pewnego alfabetu;
 - wartości liczbowe - dyskretne i ciągłe;
 - stałe, zmienne lub funkcje.



20

Operatory genetyczne: uwzględnienie specyfiki metody kodowania i umożliwienie modyfikacji:

- wartości w węzłach drzewa;
- struktury drzewa.

Obecnie:

„programowanie genetyczne” - często do określenia wszelkich algorytmów wykorzystujących drzewiastą reprezentację zadania i modyfikujących strukturę tej reprezentacji, np:

- zadanie syntezy drzewa decyzyjnego;
- projektowanie układów elektronicznych;
- regresja symboliczna;
- ...



21

Kodowanie drzewiaste:

- Chromosom jest kodowany jako drzewo, składające się z węzłów i krawędzi.
- Informacja jest zawarta w węzłach, zaś krawędzie określają wzajemne relacje pomiędzy węzłami.
- Jeśli krawędź jest skierowana od węzła A do B, to A jest nazywany nadrzędnym, B - podrzędnym.
- Węzły:
 - terminalne (nie posiadają węzłów podrzędnych);
 - pośrednie (nieterminalne).
- Istnieje dokładnie jeden węzeł, nie posiadający nadrzędnego – korzeń drzewa.

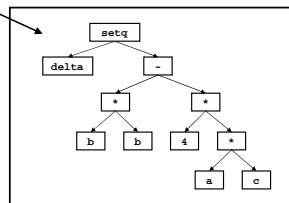


22

Przykład

Funkcja obliczająca pierwiastki rzeczywiste równania kwadratowego: $y = ax^2 + bx + c$

```
(defun pierwiastki (a b c)
  ((setq (delta -(* (b b) *(4 * (a c))))))
  (if <(delta 0)
    (setq n 0)
  )
  (if =(delta 0)
    (setq n 1)
    (setq x1 (/((-b)*(2 a))))
  )
  (if >(delta 0)
    (setq n 2)
    (setq x1 (/((-b sqrt(delta))(*
      2 a))))
    (setq x2 (/((+(-b sqrt(delta))(*
      2 a))))))
  )
  )
)
```



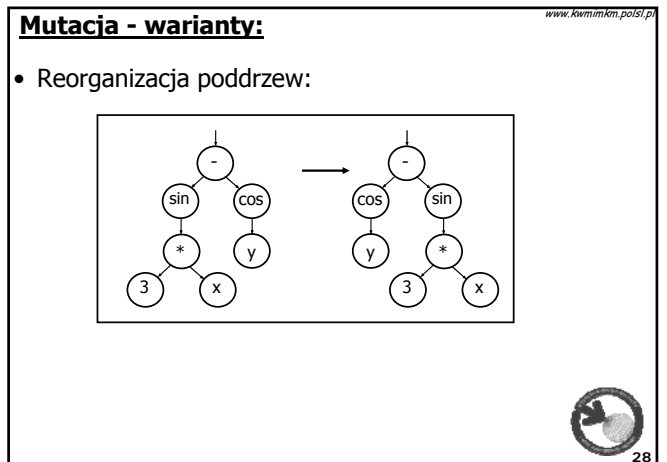
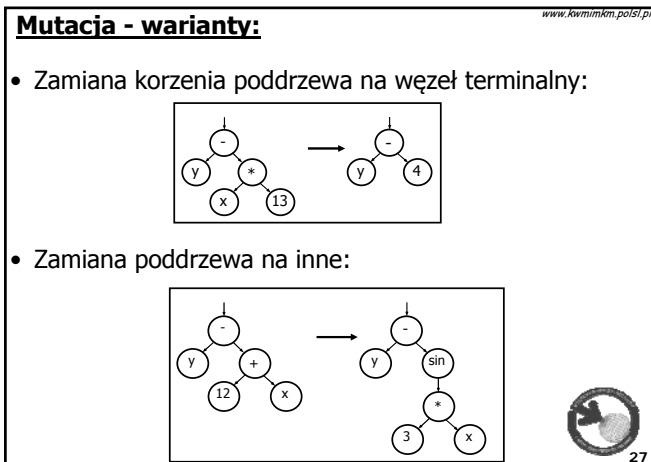
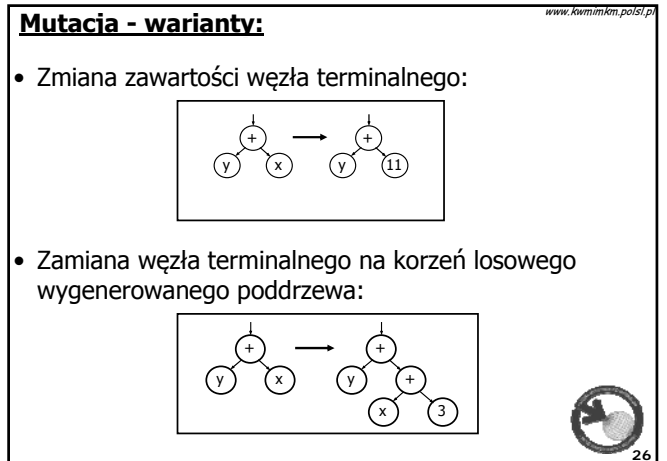
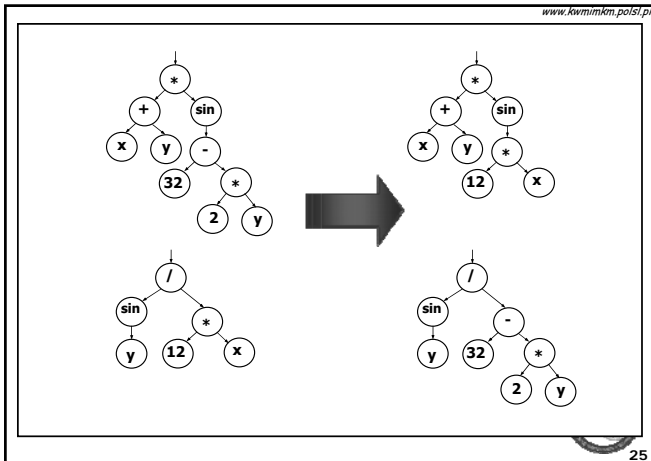
23

Krzyżowanie:

- Jest wykonywane dla pary osobników rodzicielskich i prowadzi do powstania pary osobników potomnych.
- Z każdego z osobników rodzicielskich wyodrębniany jest losowo wybrany węzeł – pośredni (wraz ze swoim poddrzewem) lub terminalny.
- Chromosomy potomne powstają w wyniku zamiany powstałych poddrzew.



24



www.kwmimkm.polisi.pl

STRATEGIE EWOLUCYJNE

29

www.kwmimkm.polisi.pl

STRATEGIA EWOLUCYJNA (1+1)

- Przetwarzany jest tylko jeden chromosom $X(t)$.
- W każdym kroku generowany jest nowy chromosom $Y(t)$ poprzez mutację $X(t)$.
- Wartości funkcji przystosowania w obu chromosomach są porównywane.
- Chromosomem $X(t+1)$ staje się ten, którego wartość funkcji przystosowania jest wyższa.

30

procedure SE (1+1)

```
begin
  t:=0
  inicjalizacja X(t)
  ocena X(t)
  while (not warunek zakończenia) do
  begin
    Y(t) := mutacja X(t)
    ocena Y(t)
    if ( Y(t) > X(t) ) then X(t+1) := Y(t)
    else X(t+1) := X(t)

    t:=t+1
  end
end
```

31

Mutacja w strategii (1+1):

- Chromosom $Y(t)$ jest generowany poprzez dodanie losowej modyfikacji (z rozkładem normalnym) do każdego genu chromosomu $X(t)$:

$$Y_i(t) = X_i(t) + \sigma \cdot k_N$$

k_N – zmienna losowa o rozkładzie normalnym $[0,1]$.

- Wartość σ określa zasięg mutacji (*większa wartość to większe perturbacje chromosomu bazowego*).
- Algorytm doboru σ powinien uwzględniać większy zasięg mutacji w początkowej fazie a mniejszy pod koniec działania SE.

n.p. reguła 1/5 sukcesów...

32

Reguła 1/5 sukcesów:

1. Jeśli przez kolejnych k generacji liczba mutacji zakończonych sukcesem przewyższa $1/5$ ogólnej liczby wykonanych mutacji, to należy zwiększyć zasięg mutacji: $\sigma = C_j \cdot \sigma$.
2. Gdy dokładnie $1/5$ mutacji kończy się sukcesem, wartość σ nie wymaga modyfikacji.
3. W przeciwnym przypadku należy zawęzić zasięg mutacji według wzoru $\sigma = C_d \cdot \sigma$.

- Wartość „ $1/5$ ” została zaproponowana na podstawie rozważań teoretycznych.
- Wartości ustalone eksperymentalnie:

$$C_d = 0.82, \quad C_j = 1/0.82$$

33

STRATEGIA EWOLUCYJNA ($\mu+\lambda$)

- Jedna z najczęściej stosowanych.
- Bardziej odporna na minima lokalne.
- Wprowadzenie mechanizmu samoczynnej adaptacji zasięgu mutacji (zastępuje regułę 1/5 sukcesów).
- Wprowadzenie operatora krzyżowania.
- Przetwarzana jest bazowa populacja $P(t)$ zawierająca μ osobników o specjalnej strukturze.
- Nową populację bazową tworzy μ najlepszych osobników wybranych spośród $\mu+\lambda$ znajdujących się w złączeniu populacji $P(t)$ i $O(t)$.

34

procedure SE ($\mu+\lambda$)

```
begin
  t:=0
  inicjalizacja P(t)
  ocena P(t)
  while (not warunek zakończenia) do
  begin
    T(t) := reprodukcja P(t)
    O(t) := krzyżowanie i mutacja T(t)
    ocena O(t)
    P(t+1) :=  $\mu$  najlepszych osobników z  $P(t) \cup O(t)$ 
    t:=t+1
  end
end
```

35

Struktura osobnika:

Dwa chromosomy:

1. Wektor X wartości zmiennych niezależnych;
2. (Zwykle) wektor σ zawierający wartości standardowych odchyłeń wykorzystywanych podczas mutacji.

Mutacja:

- Trójetapowa, związana z samoczynną adaptacją zasięgu mutacji tak, by minimalizować wartość oczekiwaną czasu dojścia do optimum globalnego.

Krzyżowanie:

- Najczęściej uśrednianie lub wymiana wartości wektorów X i σ chromosomów macierzystych – w wyniku powstają 2 chromosomy potomne.

36

STRATEGIA EWOLUCYJNA (μ, λ)

- Strategia ($\mu + \lambda$) zawodzi w przypadku, gdy w populacji znajdzie się osobnik o wyróżniającej się wartości f. przystosowania, lecz zbyt dużych lub zbyt małych wartościach standardowych odchyień.
- Różnica: nowa populacja bazowa jest tworzona wyłącznie na podstawie osobników potomnych z populacji $O(t)$.



37

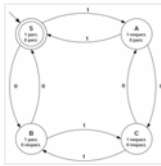
PROGRAMOWANIE EWOLUCYJNE



38

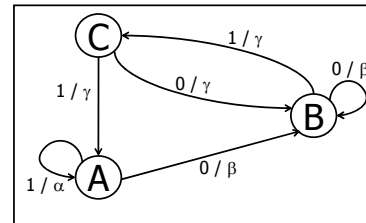
Ewoluujące automaty, L. Fogel (1966)

- Zadanie poszukiwania w przestrzeni automatów skończonych o niewielkiej liczbie stanów.
- Automat skończony (*finite state machine, FSM*) - abstrakcyjny, matematyczny, iteracyjny model zachowania systemu dynamicznego oparty o tablicę dyskretnych przejść między jego kolejnymi stanami (diagram stanów).
- Ważne narzędzie teoretyczne m. in. w tworzeniu i testowaniu oprogramowania.
- Maszyna Turinga jest generalizacją automatu skończonego operującą na nieskończonej pamięci.



39

Np.:



Jeśli automat znajduje się w stanie A, to:

- podanie na wejście „1” powoduje wypisanie na wyjściu α i pozostanie w stanie A;
- podanie na wejście „0” powoduje wypisanie na wyjściu β i przejście do stanu B;



40

Laurence Fogel (1966), David Fogel (1980')

- Odkrywana była gramatyka nieznanego języka, gdy znany był zestaw symboli języka i przykłady wyrażeń syntaktycznie poprawnych;
- Gramatyka modelowana za pomocą automatu skończonego;
- Poszukiwany zbiór stanów, funkcja przejść i funkcja wyjść.
- Funkcja przystosowania obliczana na podstawie liczby poprawnych syntaktycznie wyrażeń poznawanego języka.



41

Działanie algorytmu:

- Osobnik (D. Fogel) ma 2 chromosomy, drugi zawiera wektor odchyień stand. dla zmiennych niezależnych;
- Każdy osobnik tworzy potomka, który jest mutowany;
- Mutacja: 3 warianty:
 - 1) dodanie do genu wartości zm. losowej o rozkładzie normalnym;
 - 2) adaptacja zależna od wartości f. przystosowania;
 - 3) samoczynna adaptacja zasięgu (jak w SE);
- Selekcja: wybór osobników o najwyższej randze.
- Ranga osobnika równa l. osobników o niższym przystosowaniu.



42

```

procedure Programowanie Ewolucyjne
begin
   $t := 0$ 
  inicjalizacja  $P(t)$ 
  ocena  $P(t)$ 
  while (not warunek zakończenia) do
    begin
       $O(t) := \emptyset$ 
      foreach ( $X \in P(t)$ ) do
        begin
           $Y :=$  mutacja  $X$ 
           $O(t) := O(t) \cup \{Y\}$ 
        end
      end
      ocena  $O(t)$ 
      foreach ( $X \in P(t) \cup O(t)$ ) do
        begin
          wyznacz rangę( $X$ )
        end
      end
       $P(t+1) :=$  wybór najwyższych rangę z  $P(t) \cup O(t)$ 
       $t := t+1$ 
    end
  end

```

3

SYSTEMY (ALGORYTMY) MRÓWKOWE



44

- Mrówki są praktycznie ślepe, lecz potrafią znaleźć najkrótszą drogę do pożywienia i z powrotem.
- Ich obserwacja była inspiracją do powstania (Dorigo, 1996) nowego typu algorytmów zwanych mrówkowymi (*ant algorithms, ant systems*)

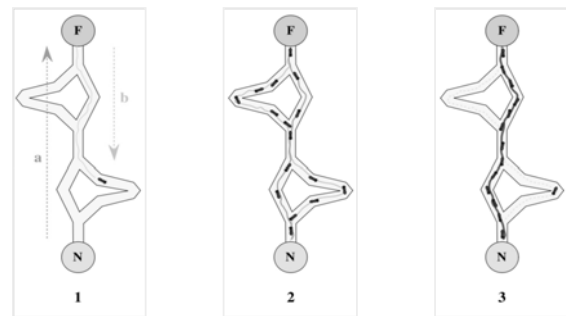


Zastosowania - zad. optymalizacji kombinatorycznej:

- problem komiwojażera;
- harmonogramowanie
- wyznaczanie tras w sieciach telekomunikacyjnych;
- wyznaczanie optymalnych tras w ruchu miejskim.



45



F – food source
N – nest

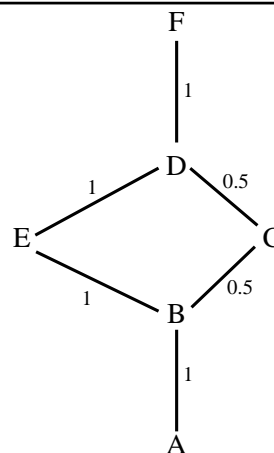


46

- Systemy bazujące na inteligencji masowej – populacja mrówek (podobnie jak w AE).
- Każda mrówka w populacji poszukuje rozwiązania (najkrótszej drogi).
- Pozostawianie feromonu na trasie.
- Wybór trasy na podstawie ilości feromonu
- Wielokrotne powtarzanie: trasa optymalna.



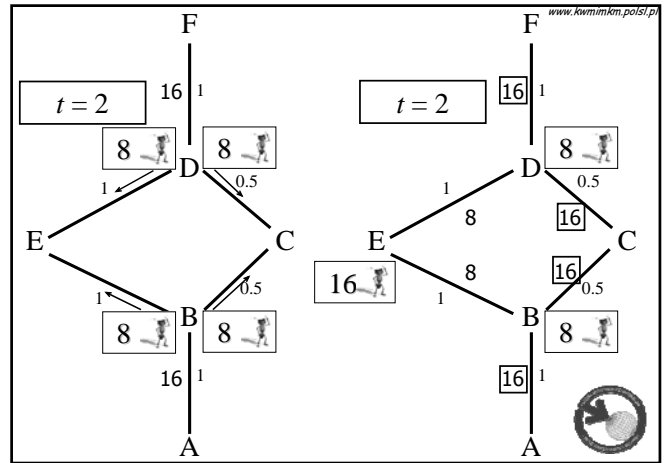
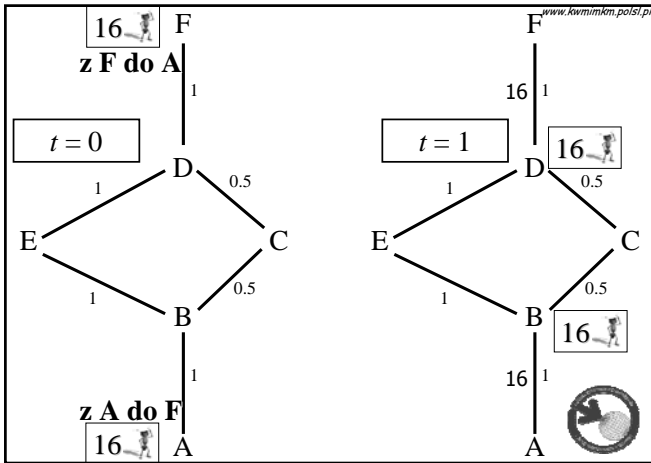
47



- Czas (t) jest dyskretny;
- W każdej jednostce czasu każda mrówka przemieszcza się o $d = 1$ pozostawiając 1 jednostkę feromonu.
- W chwili $t = 0$ brak feromonu na wszystkich krawędziach.
- Znaleźć najkrótszą drogę między A i F...



48



- www.kwmimkn.polsl.pl
- Proste podążanie ścieżką z największą ilością feromonu – szybkie utknięcie w optimum lokalnym.
 - Potrzebna jest eksploracja ścieżek – mrówki wybierają ścieżki z prawdopodobieństwem proporcjonalnym do intensywności śladu feromonowego na niej.
 - To, że mrówka wybierze daną ścieżkę zależy zarówno od intensywności feromonu jak również np. odległości od następnego miasta (TSP).
 - W celu zapewnienia braku nieograniczonego wzrostu ilości feromonu – parowanie feromonu.
- 51

www.kwmimkn.polsl.pl

<http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>

Ant Colony Optimization

NEWS

ANTS 2006: First International Workshop on Ant Colony Optimization and Swarm Intelligence, Université Libre de Bruxelles, Brussels, Belgium (Sep 4-7, 2006)

The winner of the best paper award at ANTS 2006 will receive an art designed by the Italian sculptor Matteo Pavesio

The book *Ant Colony Optimization*, by Marco Dorigo and Thomas Stützle, is available since June 2004.

The software accompanying the *Ant Colony Optimization* book is available here as open source distributed under the GPL license.

On November 22, 2005, the King of Belgium presented the "ENRS - Dr. A. Dr. Lucien Swier, Doctor in Applied Sciences" to Marco Dorigo for his fundamental contributions to the foundation of the swarm intelligence research field.

52