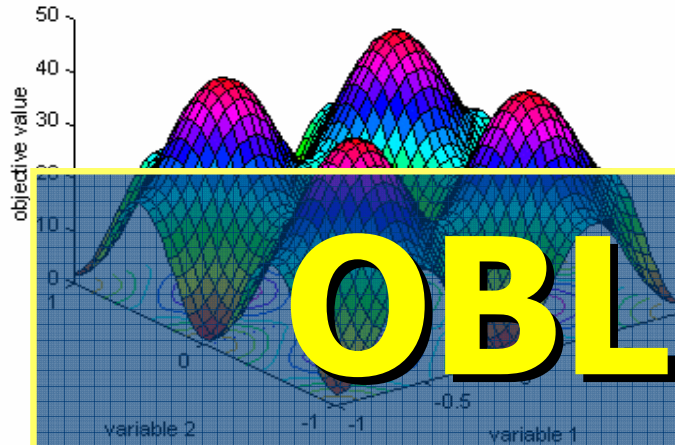
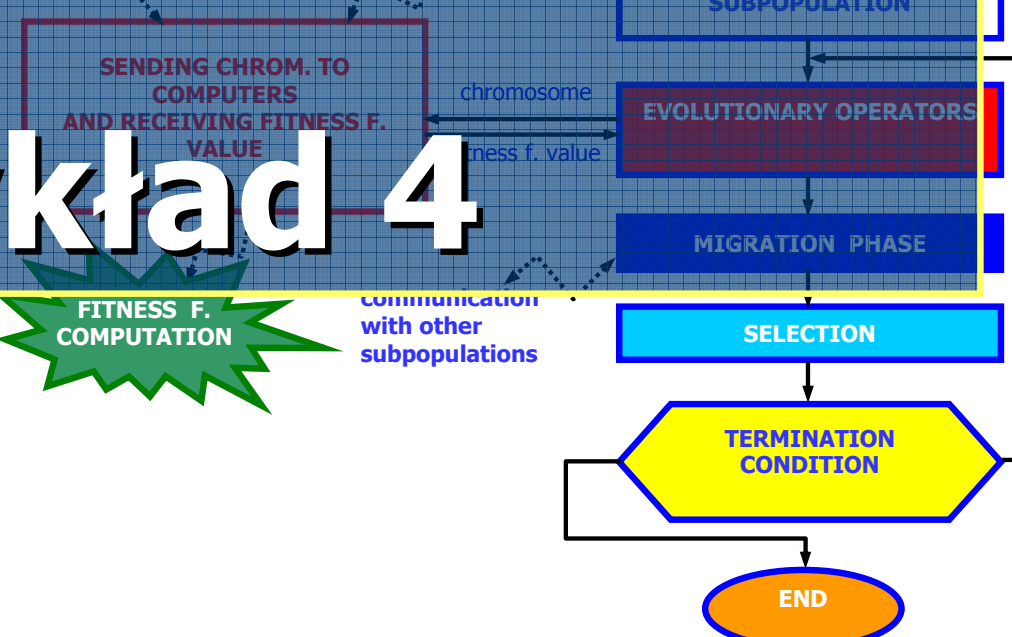
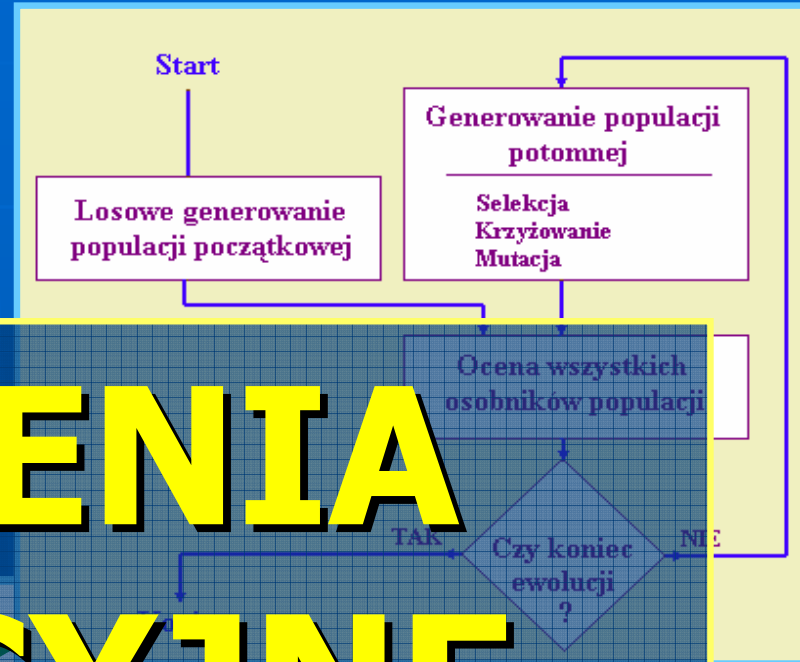


RASTRIGINs function 6



OBLICZENIA EWOLUCYJNE

wykład 4



Test: porównanie wydajności kodowania binarnego i zmiennopozycyjnego.

Zadanie sterowania:

$$J = \min \left[x_N^2 + \sum_{k=0}^{N-1} (x_k^2 + u_k^2) \right]$$

STEROWANIE OPTYMALNE - sterowanie automatyczne, w którym następuje ukształtowanie przebiegu sygnału sterującego powodujące maksymalizację lub minimalizację wartości funkcji celu.

(charakteryzującej np. wydajność produkcji albo zużycie paliwa)

Ograniczenia:

$$x_{k+1} = x_k + u_k \quad k = 0, 1, \dots, N-1$$

gdzie: x_0 – stan początkowy;

$x_k \in R$ – stan;

$u \in R^N$ – poszukiwany wektor sterowania.



Chromosom \equiv wektor sterowania \mathbf{u}

Dziedzina: $\langle -200, 200 \rangle$ dla każdego u_i .

Przyjęto: $x_0=100$, $N=45$ ($\mathbf{u} = \langle u_0, \dots, u_{44} \rangle$).

Optimum:

$$J^* = K_0 h_0^2$$

tu: $J^* = 16180.4$



Wersja binarna:

- Każdy element wektora chromosomu zakodowano za pomocą tej samej liczby bitów;
- Każdy chromosom jest wektorem składającym się z N słów;
- ☹ Nie pozwala na zwiększenie dokładności bez zwiększenia liczby bitów;
- ☹ Przy wzroście rozmiarów dziedziny dokładność maleje przy konieczności zachowania stałej liczby bitów.



Wersja zmiennopozycyjna:

- Każdy chromosom to wektor liczb zmiennopozycyjnych o długości zgodnej z wektorem rozwiązania;
 - Operatory określono tak, by każdy element chromosomu mieścił się w wymaganym zakresie.
-
- 😊 Pozwala uwzględnić bardzo duże dziedziny jak również przypadki o nieznanym dziedziny;
 - 😊 Łatwiej jest zaprojektować specjalistyczne narzędzia ułatwiające postępowanie w przypadku nietrywialnych ograniczeń.



Porównywalność algorytmów:

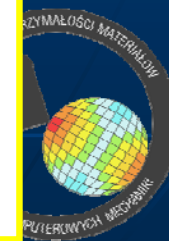
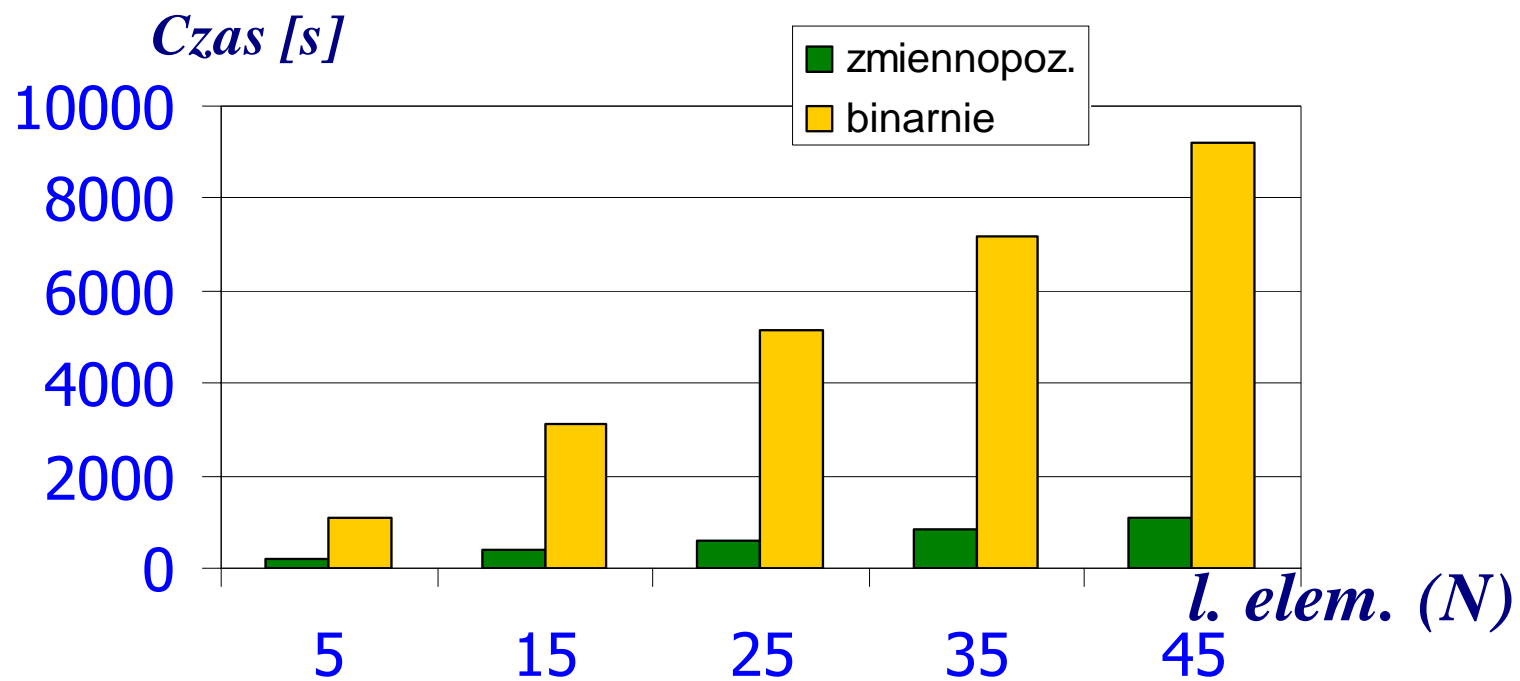
- Stała liczebność populacji (**60 osobników**)
- Stała liczba pokoleń (**20 000**)
- W reprezentacji binarnej użyto 30 bitów do zakodowania jednej zmiennej, co daje:
 $30 \cdot 45 = \mathbf{1350}$ bitów w chromosomie.
- Mimo użycia różnych operatorów *(co wynika ze sposobu kodowania zadania i może powodować różnice w interpretacji)* parametry programu dobrano tak, by wyniki mogły zostać **uczciwie porównane**.

(np. w przypadku reprezentacji binarnej użyto klasycznych operatorów, jednak zezwolono na krzyżowanie tylko pomiędzy elementami).



Wyniki:

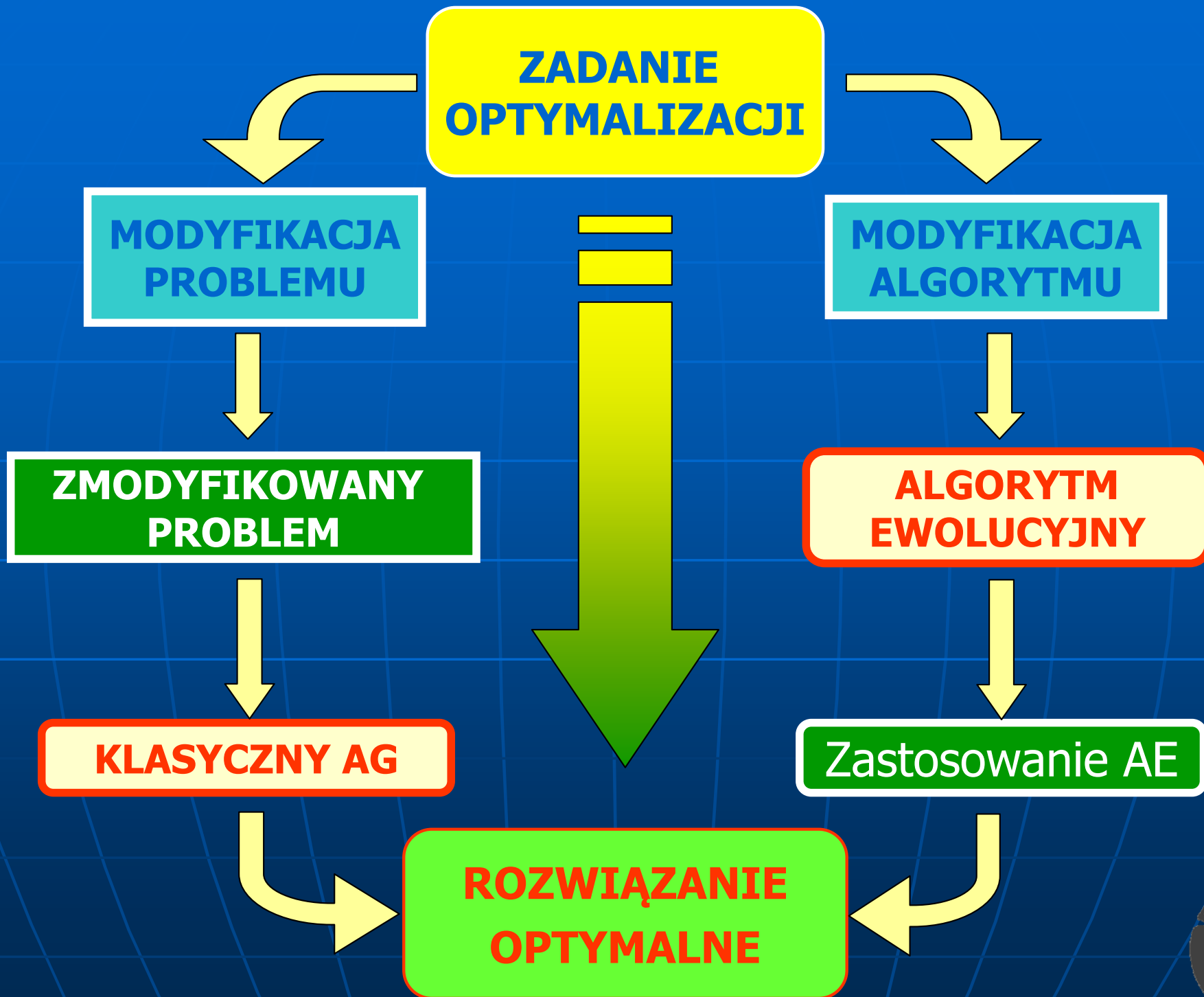
L. elementów (N)	Czas CPU [s]	
	zmiennopoz.	binarnie
5	184	1080
15	398	3123
25	611	5137
35	823	7177
45	1072	9221



Wnioski z testów:

- Reprezentacja zmiennopozycyjna jest **szybsza**.
- Reprezentacja zmiennopozycyjna jest **stabilniejsza** (daje bardziej zbliżone wyniki w różnych przebiegach).
- Reprezentacja zmiennopozycyjna jest **dokładniejsza** (szczególnie w większych dziedzinach).
- Działanie algorytmów (szybkość, zbieżność) można poprawić wprowadzając **specjalne operatory**.
- W przypadku kodowania binarnego dla dużych dziedzin i wymaganej większej dokładności różnice w czasach obliczeń **powiększają się**.





Modyfikacje:

- łańcuchy o zmiennej długości;
- struktury bogatsze od łańcuchów (np. macierze);
- zmodyfikowane operatory;
- nowe operatory (inwersja, klonowanie, itp.);
- inna niż binarna reprezentacja zadania;
- „pamięć” chromosomu;
- ...

„zmieniony AG”, „ulepszony AG”, „zmodyfikowany AG”,

...



Różnorodne programy opierające się na **zasadzie ewolucji** mogą się różnić:

- strukturą danych;
- operatorami;
- metodami tworzenia populacji początkowej;
- sposobami uwzględniania ograniczeń zadania;
- parametrami.

Zasada działania nie zmienia się:

populacja osobników podlega pewnej transformacji zaś osobniki starają się przetrwać w procesie ewolucji.

AG  **AE**



ALGORYTMY EWOLUCYJNE

Rozwinięcie idei klasycznych AG w kierunku systemów bardziej skomplikowanych, zawierających:

- odpowiednie struktury danych (kodowanie);
- odpowiednie operatory.

Słabość AE – podstawy teoretyczne:

- tylko dla czystych AG istnieje tw. o schematach;
- w innych podejściach tylko w niektórych przypadkach można wykazać teoretycznie ich zbieżność (*np. strategie ewolucyjne stosowane do zadań regularnych*).

Zwykle jednak tylko uzyskujemy **interesujące wyniki...**

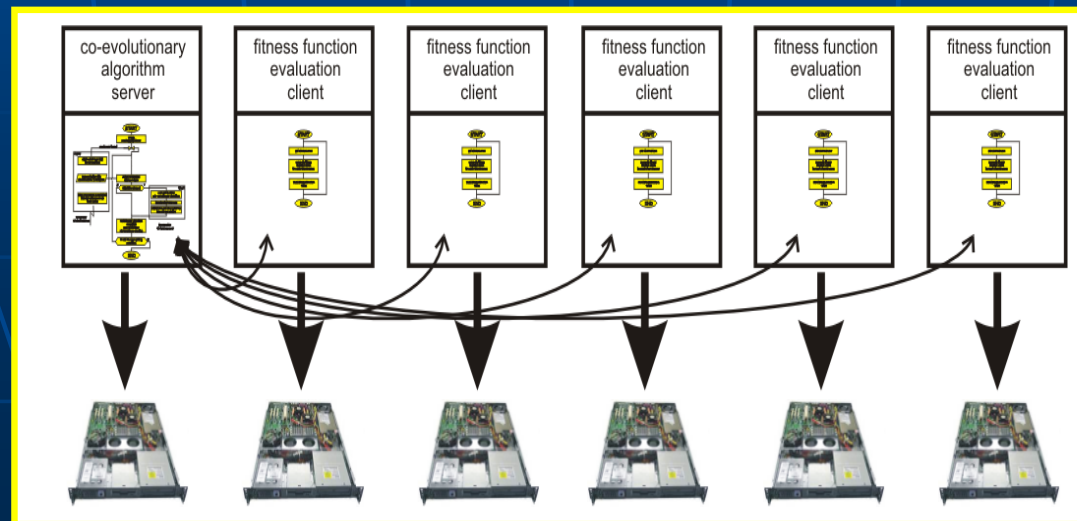


Równoległość AG i AE:

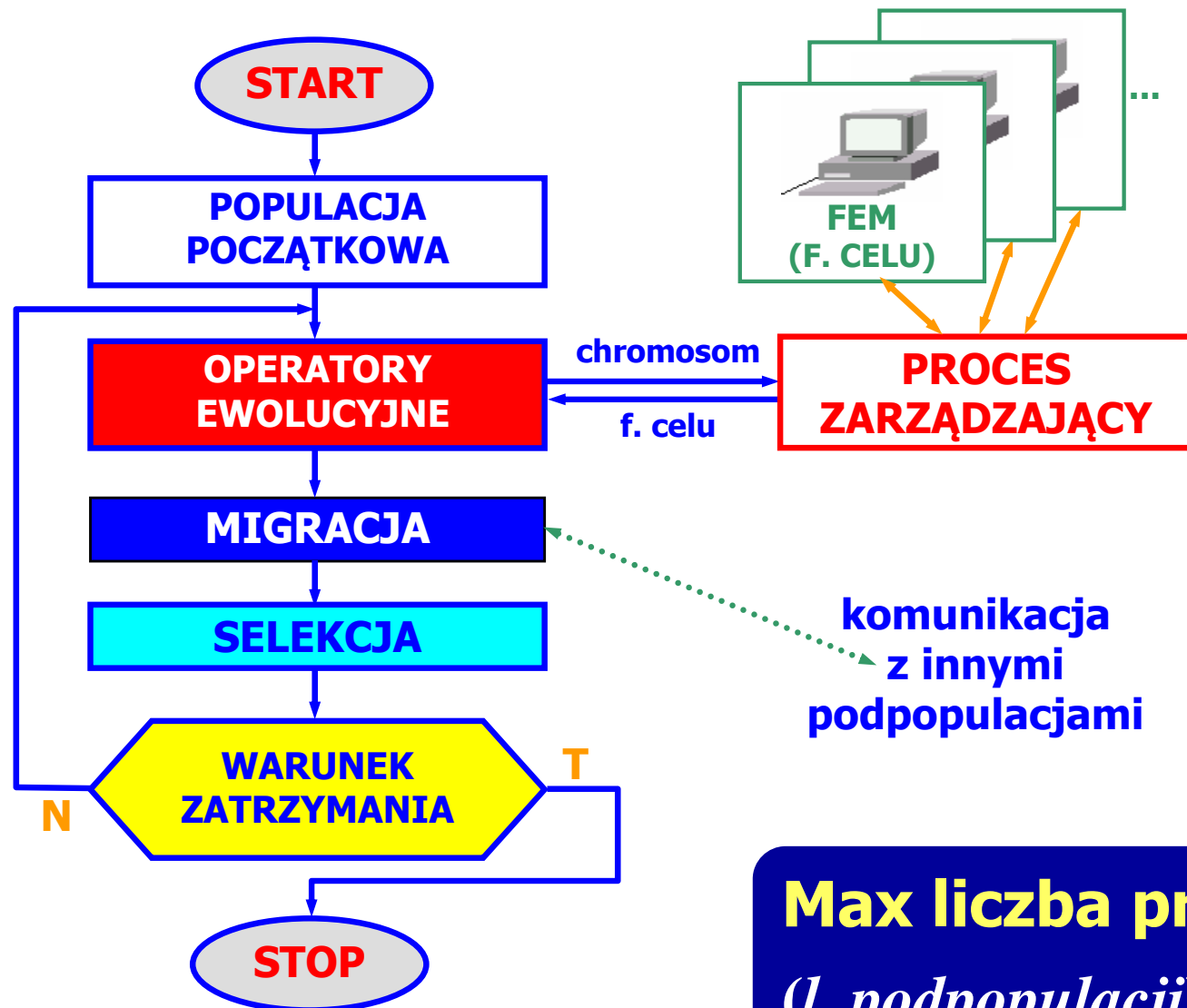
„W świecie, w którym algorytmy sekwencyjne są przerabiane na równoległe za pomocą niezliczonych sztuczek i łamańców, jest niemałą ironią, że AG (algorytmy wysoko równoległe) są przerabiane na sekwencyjne za pomocą równie nienaturalnych sztuczek i wykrętów”

Goldberg, 1995

Rozproszony AE – znaczące przyspieszenie obliczeń



Rozproszony AE, autor: Waław KUŚ:



Max liczba procesorów:

(l. podpopulacji) · (l. osobników)

Dla danego problemu można określić wiele sposobów kodowania i zdefiniować szereg operatorów (np. zadanie komiwojażera).

AE to rozwinięcie i uogólnienie AG.

Należy jednoznacznie określić:

- schemat działania AE;
- metodę selekcji;
- sposób kodowania i operatory genetyczne;
- środowisko działania AE.



procedure Algorytm_Ewolucyjny**begin** $t:=0$ wybierz populację początkową $P(t)$ ocień $P(t)$ **while** (not warunek_zakończenia) **do****begin**wybierz $T(t)$ z $P(t)$ (*reprodukcja*)utwórz $O(t)$ z $T(t)$ (*działanie operatorów ewolucyjnych*)ocień $O(t)$ utwórz $P(t+1)$ z $O(t)$ i $P(t)$ (*sukcesja*) $t:=t+1$ **end****end** T – temporary - tymczasowy O – offspring - potomny

SELEKCJA

=

REPRODUKCJA

(preselekcja)

+

SUKCESJA

(postselekcja)

procedure Algorytm_Ewolucyjny

begin

$t:=0$

wybierz populację początkową $P(t)$

ocień $P(t)$

while (not warunek_zakończenia) do

begin

wybierz $T(t)$ z $P(t)$ (reprodukcja)

utwórz $O(t)$ z $T(t)$ (operatory)

ocień $O(t)$

utwórz $P(t+1)$ z $O(t)$ i $P(t)$ (sukcesja)

$t:=t+1$

end

end

Reprodukcja – tworzenie populacji tymczasowej $T(t)$, która jest poddawana działaniu operatorów genetycznych tworząc populację potomną $O(t)$.

Sukcesja – tworzenie nowej populacji bazowej $P(t+1)$ z populacji potomnej $O(t)$ oraz starej populacji bazowej $P(t)$.



Napór selekcyjny (selektywny nacisk):

Tendencja algorytmu do poprawiania wartości średniej przystosowania.

*Algorytm charakteryzuje się tym większym naporem selekcyjnym, im większa jest **oczekiwana liczba kopii lepszego osobnika** w porównaniu z **oczekiwaną liczbą kopii osobnika gorszego**.*



Twarda (brutalna) **selekcja** – wybór do populacji potomnej i powielanie tylko **najlepszego** osobnika (metoda stochastycznego wzrostu).

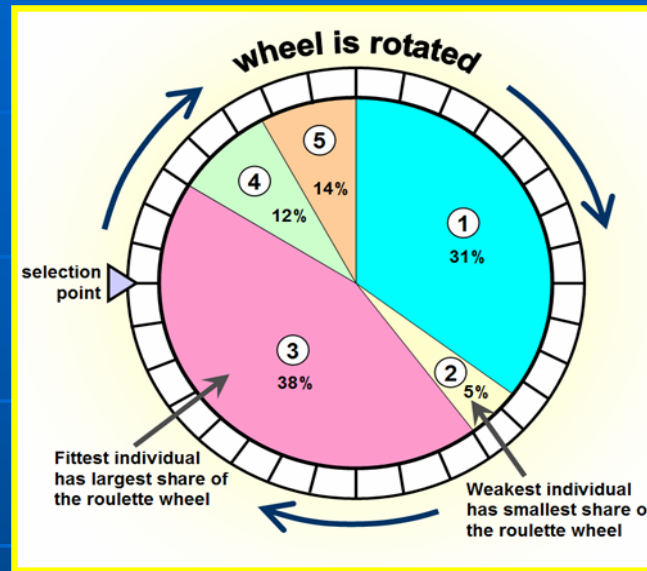
Przyjęcie **jednakowego prawdopodobieństwa** - algorytm błędzi przypadkowo (**brak selekcji**).

W AE - metoda pośrednia, zwana „**miękką selekcją**”.



METODY REPRODUKCJI

KOŁO RULETKI



— jak w AG...

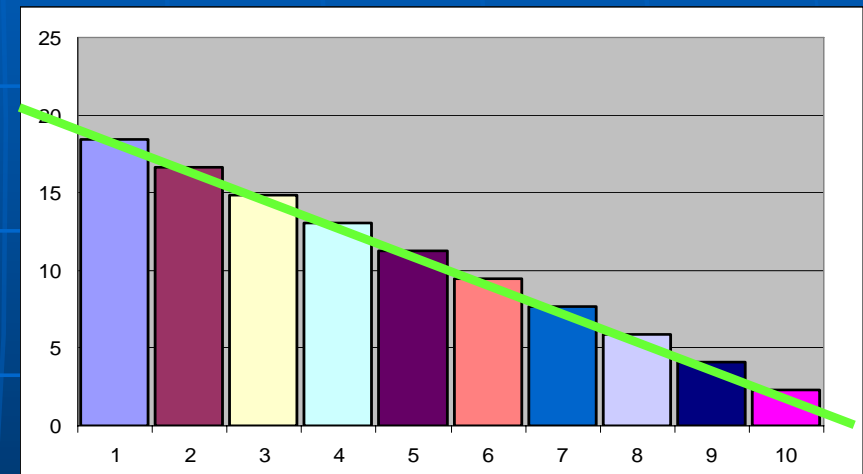
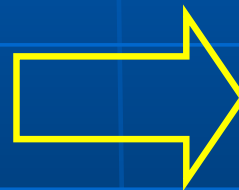
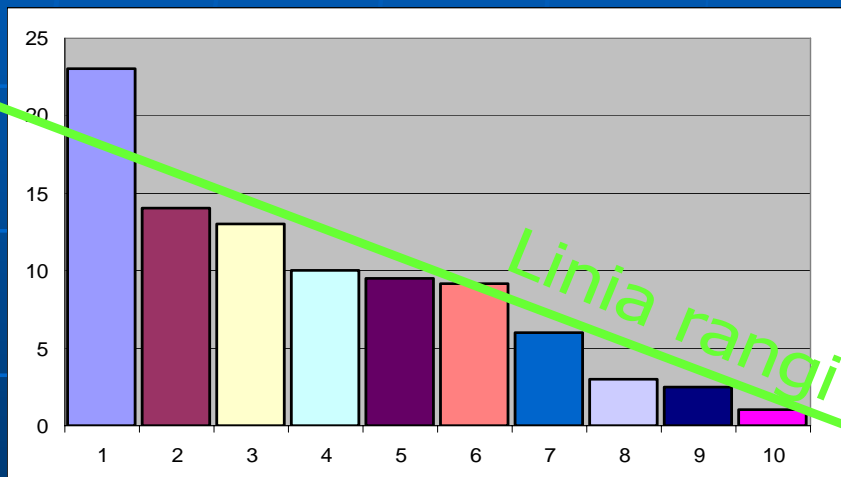
REPRODUKCJA TURNIEJOWA

- Wybór k osobników (*rozmiar turnieju, zwykle $k=2$*) i selekcja najlepszego z grupy.
- Powtarzane *pop_size* razy.



REPRODUKCJA RANKINGOWA

Szeregowanie osobników według wartości przystosowania i selekcja zgodnie z kolejnością (wg tzw. *linii rangi*):



- 😊 zapobiega powstawaniu superosobników;
- ☹ pomija informację o względnych ocenach osobników.



METODY SUKCESJI

SUKCESJA TRYWIALNA

(z całkowitym zastępowaniem)

Nową populacją bazową staje się populacja potomna:

$$P(t+1) = O(t)$$

(jak w AG).

- 😊 Najbardziej odporna na przedwczesną zbieżność.
- 😞 Najwolniej prowadzi do rozwiązania optymalnego.
- 😞 Może prowadzić do sytuacji, w której nie zawsze najlepsze rozwiązania z populacji $P(t)$ znajdą się w populacji $P(t+1)$;



SUKCESJA

Z CZĘŚCIOWYM ZASTĘPOWANIEM

W nowej populacji bazowej są osobniki z populacji potomnej i ze starej populacji bazowej

$$P(t+1) = O(t) + P(t)$$

- 😊 Prowadzi zwykle do stabilniejszej pracy AE.
- ☹️ Może spowodować tendencję do osiągnięcia maksimumów lokalnych.

Mechanizm usuwania (warianty):

- usuwanie najgorzej przystosowanych osobników;
- usuwanie osobników podobnych do potomnych;
- usuwanie losowo wybranych osobników.



SUKCESJA ELITARNA

Gwarantuje przeżycie co najmniej najlepszego osobnika poprzez odpowiedni wybór osobników z $P(t)$ do $P(t+1)$

- 😊 Wzrost wielkości elity powoduje przyspieszenie zbieżności algorytmu.
- 😞 Wzrost wielkości elity powoduje większe prawdopodobieństwo osiągnięcia ekstremów lokalnych.

Wartość **wielkości elity** δ decyduje o naporze selekcyjnym ($\delta=0$ – sukcesja trywialna).

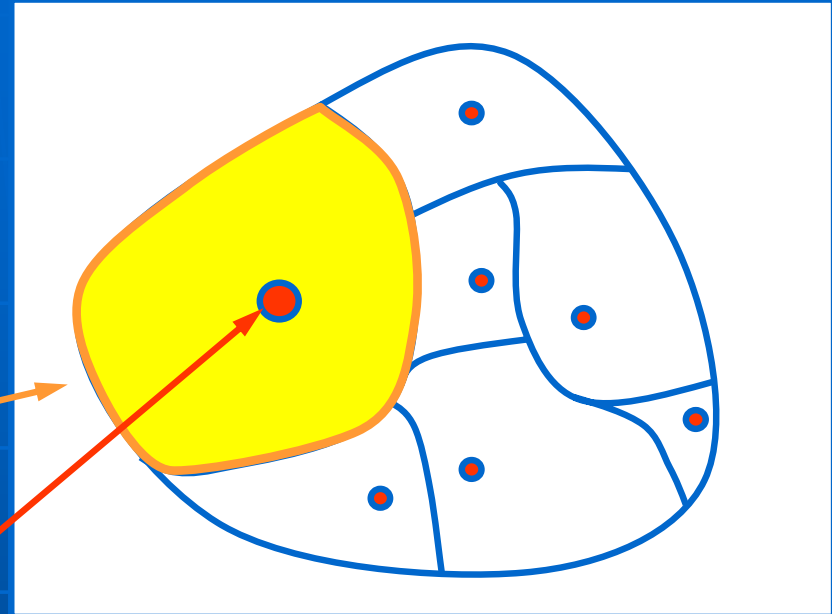
Najkorzystniej – jeden, ew. kilka osobników).



EKSPLORACJA I EKSPLOATACJA



G – dziedzina funkcji;
 \underline{X} – maksimum lokalne
 funkcji przystosowania;



Eksploracja:

Wybranie zbioru $G(\underline{X}^*)$, zawierającego maksimum globalne \underline{X}^* z rodziny obszarów przyciągania ekstremów lokalnych.

Eksploatacja:

Przeszukiwanie obszaru przyciągania $G(\underline{X})$ w celu wyznaczenie maksimum lokalnego \underline{X} (sprowadza się do zadania optymalizacji lokalnej).



- Najczęściej informacja o rodzinie obszarów przyciągania i wartości funkcji przystosowania w maksimach lokalnych **nie jest dostępna**, a jej pozyskanie jest **bardzo kosztowne**.
- Zadanie optymalizacji globalnej jest **nierozwiązywalne** (w ogólnym przypadku, w dokładnym sensie).
- Poszukuje się metod optymalizacji prowadzących do uzyskania dobrych rozwiązań **przybliżonych** z **akceptowalnym kosztem** – jak algorytmy ewolucyjne.



OCENA DZIAŁAŃ AE



LOSOWOŚĆ W AE

Różne zachowanie algorytmu w niezależnych uruchomieniach przy jednakowych ustawieniach parametrów i identycznych populacjach początkowych...

Losowość jest wprowadzana w AE:

- Podczas generowania populacji początkowej.
- W procesie wyboru populacji potomnej na drodze reprodukcji (ukierunkowanie działania AE).
- Podczas działania operatorów ewolucyjnych (próbkiwanie przestrzeni roboczej).

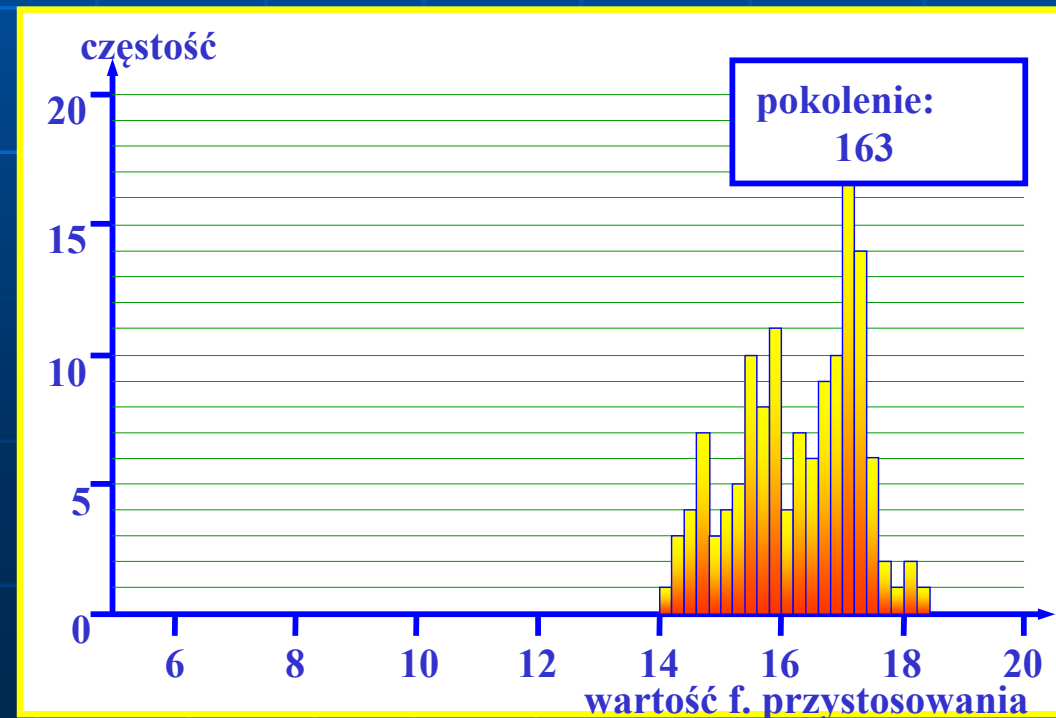


- Należy dokonać **wielu niezależnych uruchomień** dla losowej próby różnych populacji bazowych $P(0)$.
- W przypadku wielu uruchomień dla tej samej populacji $P(0)$ można mówić o właściwościach danego algorytmu **dla konkretnej populacji początkowej**.
- Porównywane dwa algorytmy: wskazane jest aby próba ta była taka sama (dla każdej losowej populacji początkowej uruchamia się dwa porównywane algorytmy).



ANALIZA STATYSTYCZNA:

- Analiza **wartości oczekiwanej** i **odchylenia standardowego** wartości przystosowania w populacji bazowej.
- Uwzględnienie informacji o min. i max. osiągniętej wartości.
- Uwzględnienie informacji o liczbie przypadków niewiele różnych od najlepszego.
- Prezentacja wyników w postaci **histogramu** pozwala ocenić właściwości rozkładu, które mogą umknąć przy analizie ograniczonej jedynie do statystyki liczbowych.

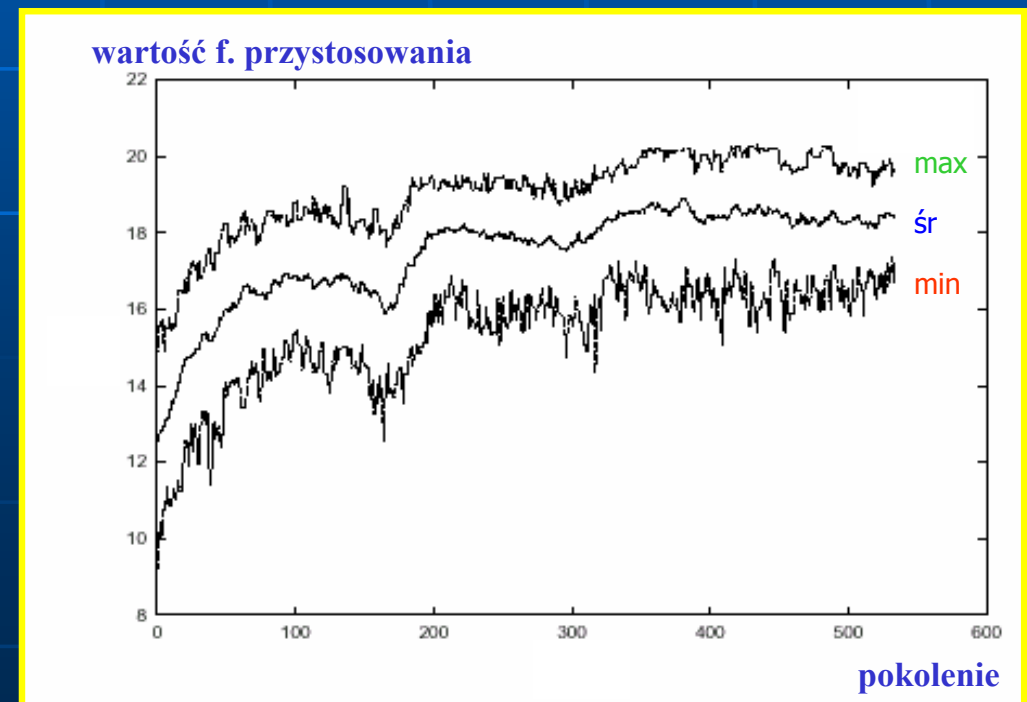


KRZYWE ZBIEŻNOŚCI

- Są wykresem zmian wartości rozwiązania roboczego (średniego, najlepszego, najgorszego) w czasie.
- Kreśli się je:
 - dla pojedynczego uruchomienia algorytmu;
 - dla wielu niezależnych uruchomień.

(Uśrednione – bardziej reprezentatywne, lecz gubi się niektóre informacje o zachowaniu AE w pojedynczych uruchomieniach).

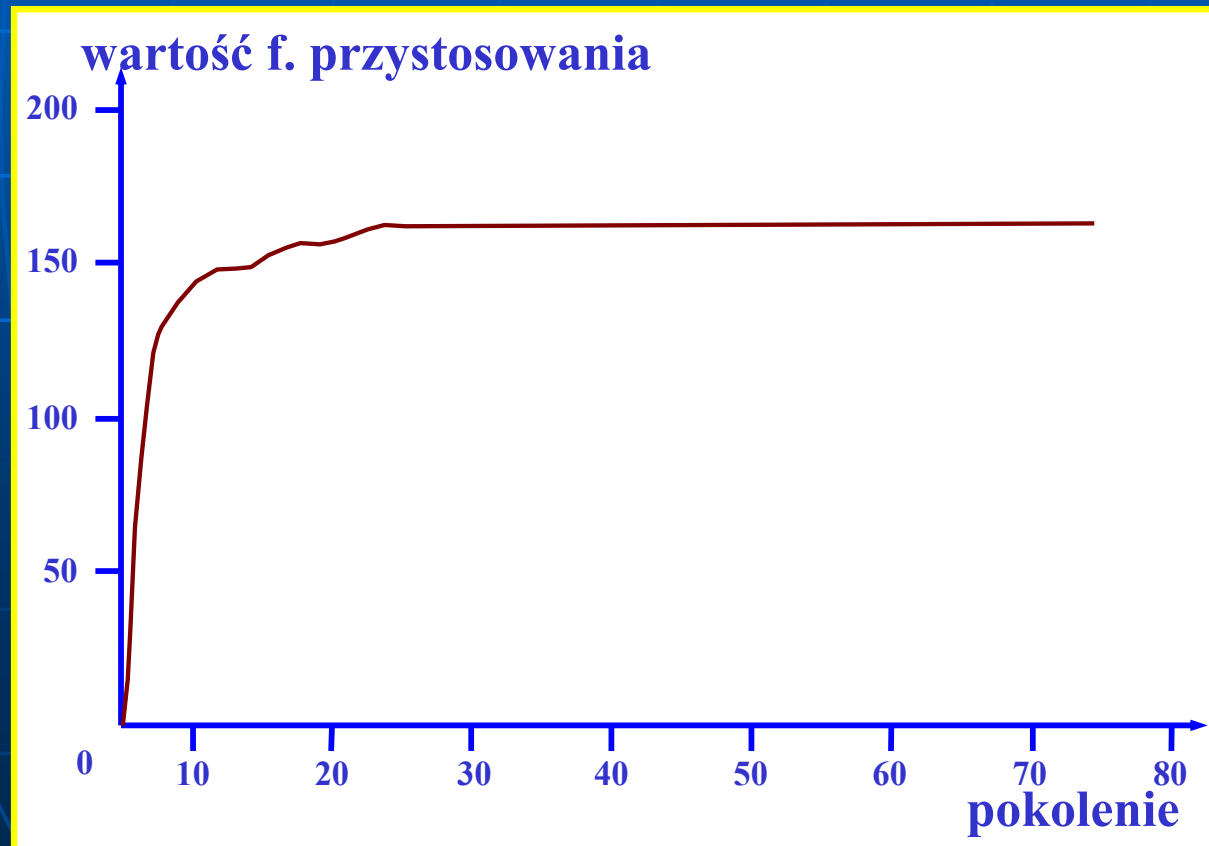
Przykładowe krzywe zbieżności
(dla 1 uruchomienia algorytmu):



Szczególna krzywa zbieżności:

Wykres zmian w kolejnych pokoleniach wartości przystosowania **najlepszego** osobnika znalezioneego **od początku** działania AE.

Po zakończeniu działania AE osobnik ten **jest rozwiązaniem** wyznaczonym. przez pojedyncze uruchomienie AE.



KOSZT SYMULACJI A LICZBA POKOLEŃ

- W wielu metodach optymalizacji **koszt** jest utożsamiany z **liczbą iteracji** metody (*przetwarzane jest jedno rozwiązanie*).
- W przypadku algorytmów ewolucyjnych liczba obliczeń wartości funkcji przystosowania w pojedynczej iteracji algorytmu ewolucyjnego jest **zależna od liczebności populacji potomnej** (przetwarzana jest populacja rozwiązań).

Np.: **100** iteracji (pokoleń) i **10** osobników
100 iteracji (pokoleń) **1000** osobników



KRYTERIA ZATRZYMANIA AE



- **Dowody zbieżności** są znane jedynie dla pewnych **szczególnych przypadków** funkcji przystosowania (takich jak funkcja kwadratowa).
- **Optymalne kryterium** - stwierdzenie, że rozwiązanie wygenerowane przez AE jest **dostatecznie dobrym przybliżeniem** maksimum globalnego funkcji celu.
- To implikuje **znajomość tego maksimum** jeszcze **przed rozpoczęciem optymalizacji...**

KRYTERIA ZATRZYMANIA:

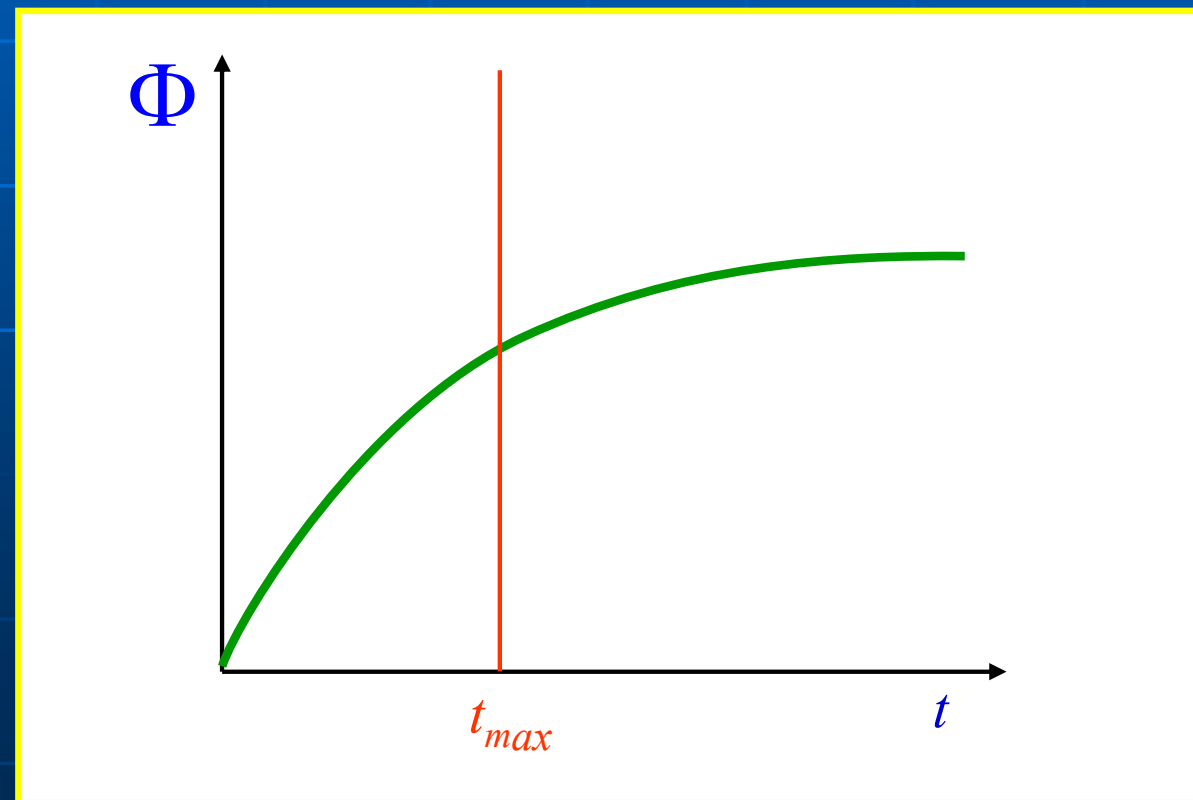
- Polegające na monitorowaniu **wartości funkcji przystosowania** osobników generowanych przez algorytm;
- Monitorujące zdolność algorytmu do **eksploracji przestrzeni genotypów**.



MONITOROWANIE WARTOŚCI FUNKCJI PRZYSTOSOWANIA

Kryterium maksymalnego kosztu

- Algorytm kończy działanie, jeśli koszt algorytmu przekroczy założoną wartość maksymalną K_{max} .
- Wartość K_{max} jest najczęściej związana ze specyfiką zadania.
- **Często**: przyjęcie pewnej maksymalnej liczby pokoleń algorytmu.



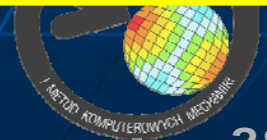
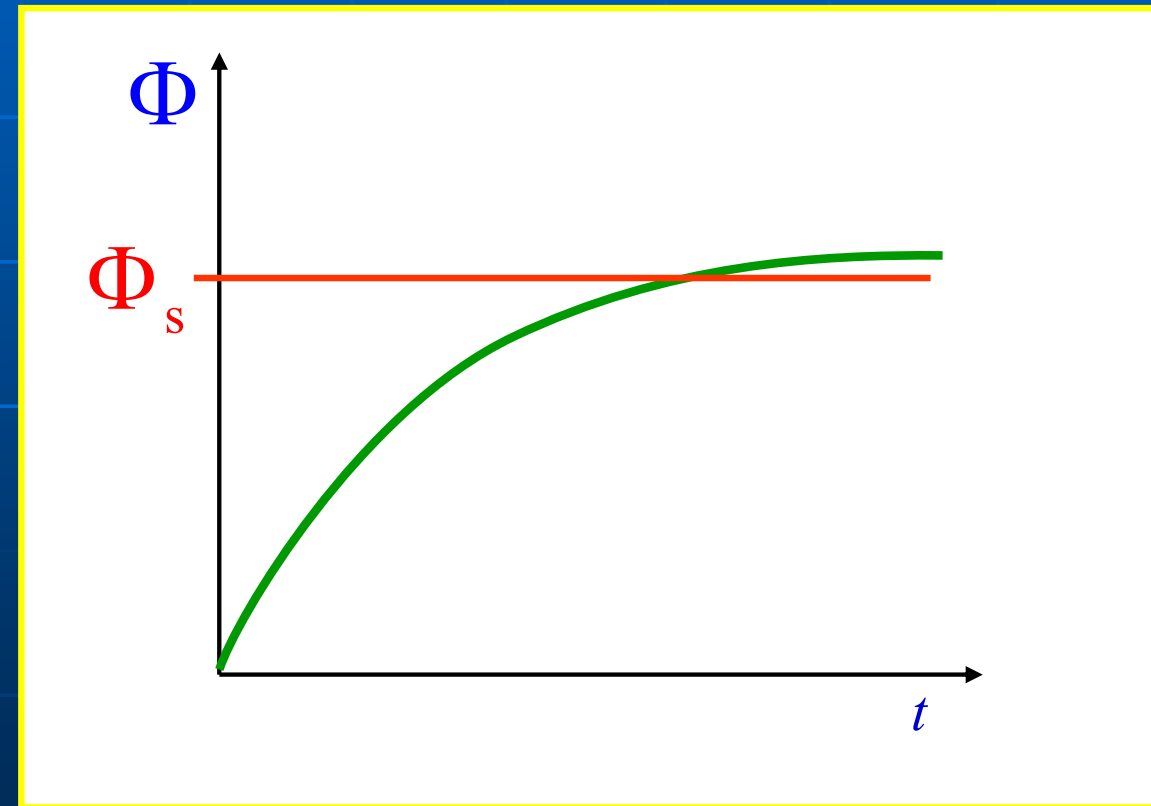
(Czas działania zależy niejawnie od liczby nowych osobników generowanych w każdym pokoleniu!)

Kryterium zadowalającego poziomu funkcji przystosowania

Zatrzymanie działania gdy AE znajdzie rozwiązanie o wartości funkcji przystosowania Φ_s określonej przez użytkownika jako zadowalająca.

☹️ Zwykle nie jest łatwo (bez dostatecznie dobrej znajomości funkcji przystosowania) **określić** wartość zadowalającą.

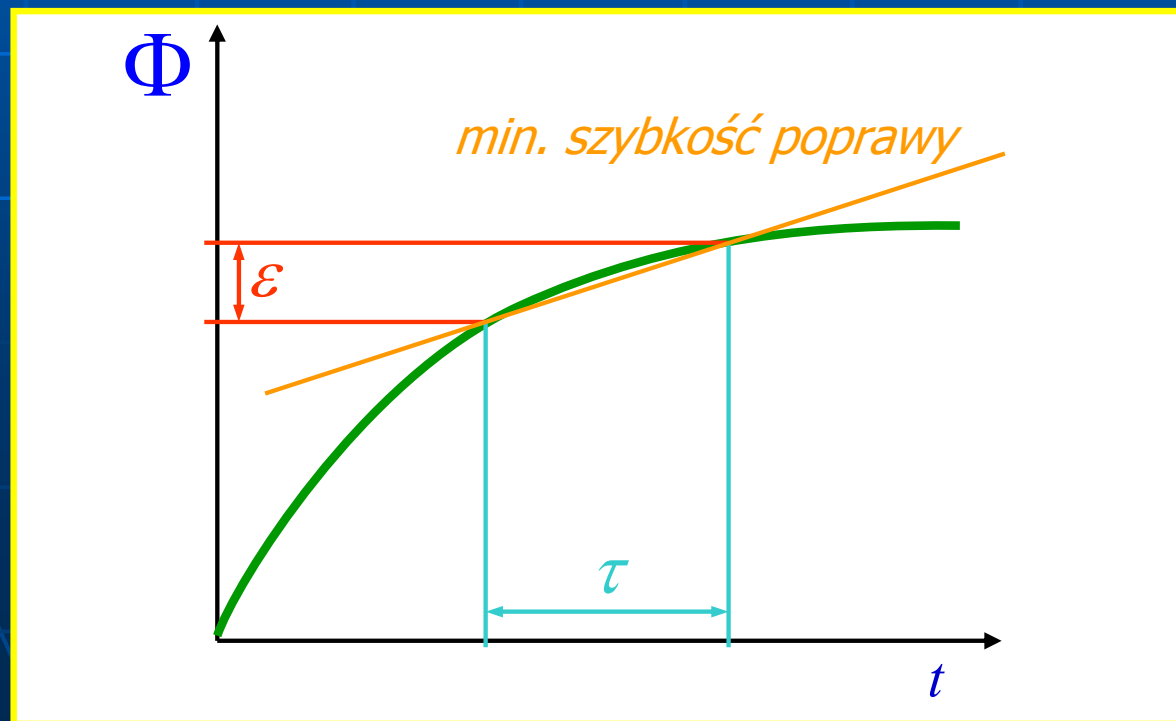
☹️ AE może działać **dowolnie długo** (należy dodatkowo określić maksymalny koszt znalezienia rozwiązania).



Kryterium min. szybkości poprawy

Algorytm jest zatrzymywany, jeśli w kolejnych τ obliczeniach wartości f . przystosowania nie uda się poprawić wyniku o więcej niż ε .

Często $\varepsilon = 0$ - AE zatrzymywany, jeśli nie uda się uzyskać lepszego rozwiązania w kolejnych τ pokoleniach.



MONITOROWANIE ZDOLNOŚCI EKSPLORACYJNYCH

- Zdolność algorytmu do eksploracji przestrzeni genotypów jest czynnikiem warunkującym odporność algorytmu na ekstrema lokalne.
- Zdolność taka wynika zarówno z **różnorodności populacji bazowej**, jak również z **zasięgu operatora mutacji** (jeśli podlega on samoczynnej adaptacji).



Kryterium zaniku różnorodności populacji

Różnorodność:

- wpływa na odporność algorytmu ewolucyjnego na ekstrema lokalne;
- sprawia, że krzyżowanie ma duży zasięg;
- powoduje w efekcie, że algorytm dość szybko generuje nowe rozwiązania.

Zanik różnorodności – przełączenie się na przeszukiwania ograniczone do niewielkiego obszaru

(zwykle równoważne z eksploatacją obszaru przyciągania jednego ekstremum – **niekoniecznie globalnego**).



Kryterium zatrzymania bazujące na różnorodności populacji wykorzystuje fakt, że **obniżenie różnorodności** poniżej pewnego poziomu świadczy o przejściu do etapu **eksploatacji obszaru przyciągania ekstremum**.

Algorytm należy wówczas **zatrzymać** i wykorzystać pewną **metodę lokalną** do dokładnego wyznaczenia tego ekstremum.



Kryterium zaniku samoczynnie adaptowanego zasięgu operatora mutacji

- Hipoteza (eksperymentalnie potwierdzona), że jeśli w AE stosuje się **adaptację zasięgu mutacji**, to od pewnego momentu zasięg ten ma trwałą **tendencję do zmniejszania się**.
- Ograniczenie zasięgu wiąże się z przełączeniem AE na **eksploatację** znalezionej przestrzeni przyciągania.
- Oblicza się dla populacji bazowej **wartość średnią standardowych odchyłeń** pamiętanych w genotypach osobników wykorzystywanych podczas mutacji.
- Spadek tej wartości poniżej pewnego progu σ_{min} – zakończenie działania.



KODOWANIE I OPERATORY GENETYCZNE



DEDYKOWANE OPERATORY GENETYCZNE

- Pozwalają **poprawić** działanie algorytmu dla pewnej **konkretnej** klasy zadań (*ściśle zwi zek z rozpatrywanym problemem*);
- Powsta le z ich u yciem AE staj  si  nierzadko **konkurencj ** wobec **innych metod rozwi zywania** tych zada ;
- Cena – **utrata og lno ci** algorytmu (*dla zada  innej klasy mo e by  on bardzo nieefektywny lub niemo liwy do zastosowania*).

Dyskusja o **wy szo ci** jednych operator w nad innymi sprowadza si  najcz ściej do przeprowadzenia **eksperyment w por wnawczych** i analizy ich wyników.



DOBÓR SPOSOBU KODOWANIA

Jeśli środowisko jest zdefiniowane na podstawie funkcji celu i ograniczeń, to wskazane jest dobranie takiego kodu genetycznego, by:

- **każde rozwiązanie** zadania dało się reprezentować jako **genotyp**;
- **kodowanie** nie wprowadzało dodatkowych **ekstremów lokalnych** funkcji przystosowania.

DOBÓR OPERATORÓW GENETYCZNYCH

1. **Spójność** algorytmu ze względu na operacje genetyczne.
2. Brak **obciążeń** operatorów.



SPÓJNOŚĆ ALGORYTMU ZE WZGLĘDU NA OPERACJE GENETYCZNE

- Jest to możliwość wygenerowania dowolnego chromosomu z dowolnego innego, posługując się **jedynie operatorami genetycznymi**.
- Spełnienie postulatu spójności wyklucza niebezpieczeństwo, że w wyniku niewłaściwej inicjalizacji populacji bazowej algorytm ewolucyjny **nie ma możliwości** znalezienia optimum globalnego.



Przykłady:

- AG z kodowaniem binarnym, krzyżowaniem jednopunktowym i bez operatora mutacji **nie spełnia** postulatu spójności
(gdy populacja bazowa zawiera kopie tego samego osobnika, to niemożliwe jest wygenerowanie jakiegokolwiek innego osobnika).
- jeśli jedynym operatorem genetycznym jest **mutacja bitowa**, to postulat spójności jest spełniony.

BRAK OBCIĄŻEŃ OPERATORÓW

Operator jest **obciążony**, jeżeli przy braku naporu selekcyjnego niektóre osobniki będą bardziej **uprzywilejowane** w stosunku do innych.

