Metody Rozmyte i Algorytmy Ewolucyjne

mgr inż. Piotr Kaczyński

Wydział Matematyczno-Przyrodniczy
Szkoła Nauk Ścisłych
Uniwersytet Kardynała Stefana Wyszyńskiego

Uwzględnianie ograniczeń
No free lunch theorem

Plan wykładu

Uwzględnianie ograniczeń
- Modyfikacja operatorów
- Algorytmy naprawy

No free lunch theorem

Przykłady zastosowań
- Adaptacyjne regulatory PID
- Regulacja predykcyjna
- Problem przykładowy

Przypomnienie

- Wykorzystanie optymalizacji lokalnej
  - Efekt Baldwina,
  - Ewolucja lamarkowska,
- Rówoległość w algorytmach ewolucyjnych
  - Zwykłe implementacje równoległe,
  - Algorytmy koewolucyjne,
- Wpływanie na czas życia osobników,
  - Limitowanie czasu życia,
  - Nacisk selektywny sterowany czasem życia,

Uwzględnianie ograniczeń

- W programowaniu nieliniowym

$$\min_{x} f(x)$$
$$g_i(x) \leqslant 0$$
$$g_e(x) = 0$$

- Występują praktycznie w każdym zagadnieniu praktycznym,
- Metody funkcji kary
  - Zewnętrzna funkcja kary,
  - Wewnętrzna funkcja kary,
- Modyfikacja algorytmu ewolucyjnego
  - Modyfikacja operatorów genetycznych i kodowania,
  - Algorytmy naprawy,

Modyfikacja operatorów genetycznych

- Generowanie populacji bazowej,
  - Losowanie tylko na obszarze dopuszczalnym,
- Specjalizowane operatory genetyczne,
  - Nie mogą wyprowadzać osobnika poza obszar dopuszczalny,
- Metoda trudna szczególnie dla ograniczeń nieliniowych,
- Trudność w utrzymaniu braku obciążenia operatorów,
- Możliwe zastosowanie dla ograniczeń równościowych,

Przykład

Przykład

Rozważmy zadanie, gdzie chromosom zawiera *n* genów, liczb całkowitych dodatnich. Dane jest ograniczenie

$$\sum_{i=1}^{n} X_i = d, \quad 0 \leqslant X_i \leqslant d$$

Metody inicjalizacji

- Losowanie genu, który przyjmie wartość *d*, pozostałe 0,
- Losowanie d razy genu i zwiększenie go o 1,

Operator mutacji

- Losowanie dwóch genów i liczby (mniejszej od mniejszego z nich); odjęcie i dodanie liczby od tych genów,

Algorytmy naprawy

- Nie są modyfikowane operatory genetyczne ani kodowanie,
- Algorytm genetyczny projektowany jest w sposób standardowy,
- Występuje dodatkowy krok „naprawiający” zły chromosom,
- „Naprawa” występuje jeśli po którejś z operacji chromosom przestał spełniać ograniczenia,
- Możliwe dwa podejścia
  - Naprawiony chromosom leży na brzegu obszaru dopuszczalnego,
  - Możliwe jest to, że naprawiony chromosom będzie leżał zarówno na brzegu, jak i w środku obszaru

Naprawa do brzegu zbioru

- Punkt niedopuszczalny jest ściągany na brzeg zbioru,
- Najczęściej wybierany jest punkt najbliższy niedopuszczalnemu,
- Zaletą jest względna prostota rozwiązania,
- Wadą fakt, że intensywniej przeszukiwane są brzegi zbioru,

Przykład

Dla ograniczeń kostkowych

$$0 \leqslant X_i \leqslant d_i$$

jeśli dane X_i nie spełnia ograniczenia, to podstawiane jest $X_i = 0$ bądź $X_i = d_i$

Naprawa do środka zbioru

- Punkt niedopuszczalny jest ściągany na brzeg lub do środka zbioru rozwiązań dopuszczalnych,
- Metoda wymagająca znajomości problemu, często trudniejsza,
- Nie wpływa na intensywność przeszukiwania brzegów,

Przykład

Problem plecakowy. Jeśli plecak jest za bardzo „załadowany” usuwane są przedmioty, aż do spełnienia ograniczenia. Kolejność usuwania może być heurystyczna.

No free lunch theorem

- Nikomu nie należy się darmowy lunch!
- Czyli: nie ma nic za darmo!
- Każdy algorytm w sensie średnim będzie miał średnio tą samą efektywność co dowolny inny jeśli uwzględni się wszystkie możliwe optymalizowane funkcje,

$$\sum_f P(h_m^y | f, m, a_1) = \sum_f P(h_m^y | f, m, a_2)$$

gdzie f funkcja celu, m ilość obserwowanych rozwiązań, a_1, a_2 dwa algorytmy, h_m^y multizbiór przeszukanych rozwiązań,

NFL - przykład

Gra w 100 ruletek

Losujemy liczby przy pomocy 100 ruletek i zakrywamy stoły. Zadaniem jest znalezienie największej wylosowanej liczby. Funkcją celu jest $100-$ ilość okrytych ruletek. Nie ma strategii optymalnej.
Jeśli przeprowadzimy bardzo dużo takich eksperymentów, to niezależnie od obranej strategii (niezależnie od gracza, czy algorytmu) odkrywania ruletek średnia efektywność będzie taka sama.

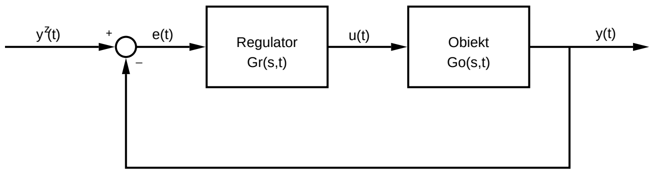
Innymi słowy — nie ma algorytmu „dobrego na wszystko”. Należy wykorzystać jak najwięcej wiedzy o problemie.

NFL a algorytmy genetyczne

- Wniosek: algorytm genetyczny nie jest i nigdy nie będzie „lekarstwem na wszystko”,
- Wiedza o problemie przejawia się przez odpowiedni dobór
  - Kodowania,
  - Operatorów genetycznych,
  - Parametrów algorytmu,
- Nie istnieje idealna strategia o idealnych parametrach dobra dla każdej funkcji celu,
- Rozwiązanie każdego problemu, nawet przy pomocy algorytmu ewolucyjnego, wymaga wnikliwej jego analizy i znajomości,
- Niestety, na lunch trzeba sobie zapracować!

Regulacja automatyczna

- Dziedzina matematyki zajmująca się sterowaniem obiektami,
- Zastosowania praktycznie wszędzie (np. tempomaty),
- Dany jest obiekt sterowany (np. przez równanie różniczkowe) i regulator



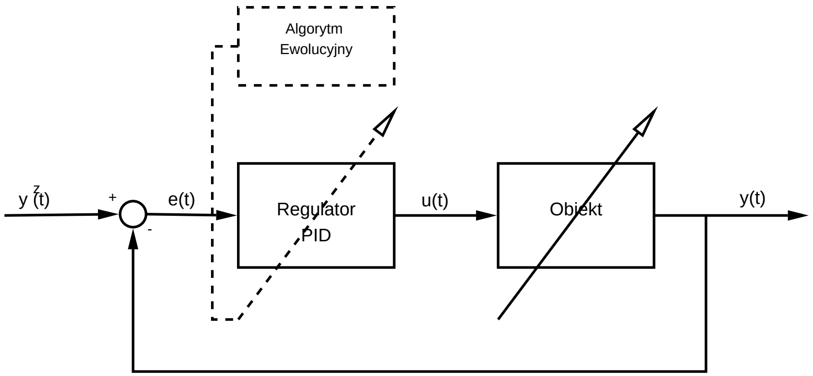
Regulator PID

- Najprostszy i najczęściej stosowany regulator,
- Regulator liniowy do obiektów liniowych,
- Sterowanie wyznaczane na podstawie uchybu ε,
- Charakteryzowany tylko trzema parametrami wzmocnieniem (K), czasem zdwojenia (Ti) i czasem wyprzedzenia (T_d),

$$U(t) = K_p \left[\varepsilon(t) + \frac{1}{T_i} \int \varepsilon(t) dt + T_d \frac{d\varepsilon(t)}{dt} \right]$$

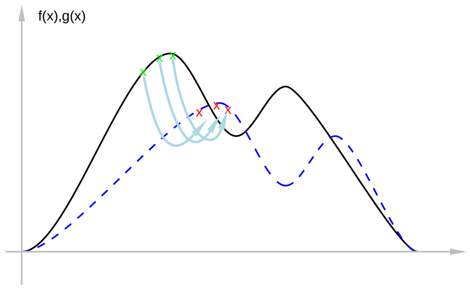
- Parametry te można dobrać eksperymentalnie (Reguły Zieglera-Nicholsa),

Regulator PID i adaptacyjny algorytm ewolucyjny

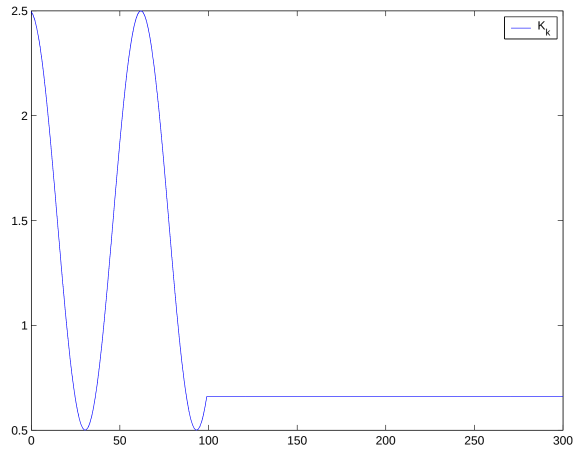


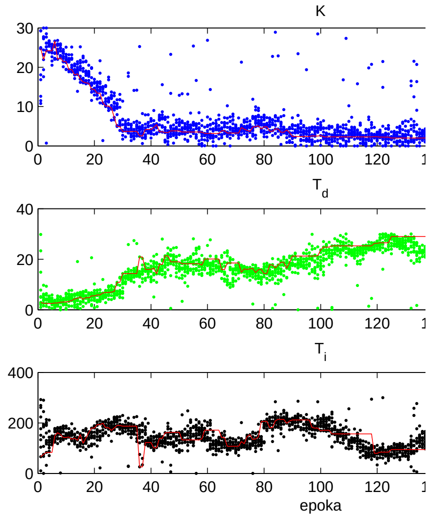
Regulator PID i adaptacyjny algorytm ewolucyjny II

- Obiekt jest niestacjonarny,
- Parametry regulatora muszą być strojone w trakcie działania układu,
- Strojenie przy pomocy algorytmu ewolucyjnego,
- Algorytm ewolucyjny działa cały czas (1 sekunda czasu rzeczywistego = 1 epoka),

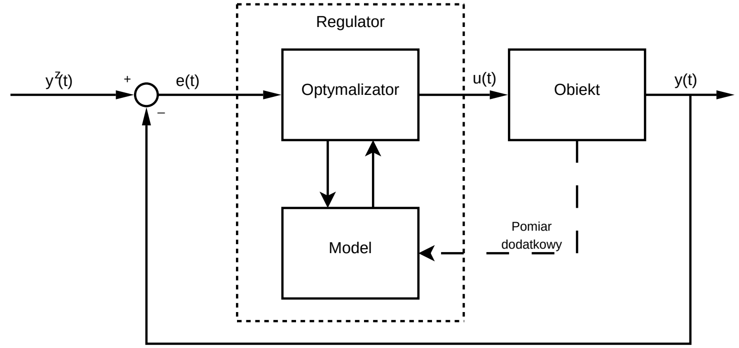


Wyniki symulacyjne

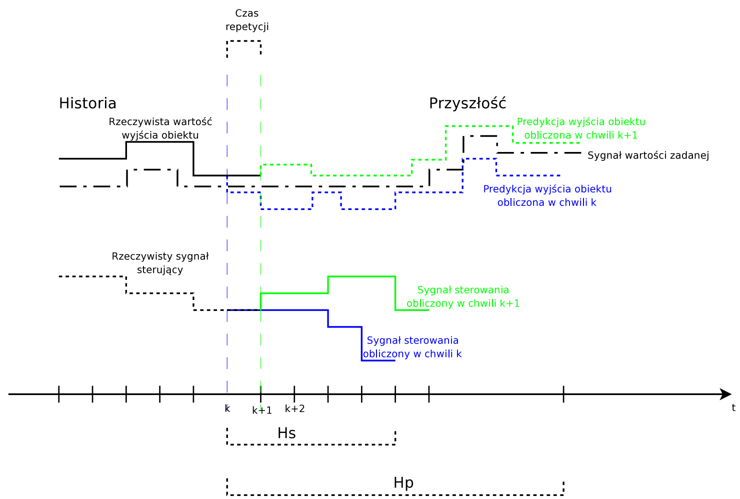




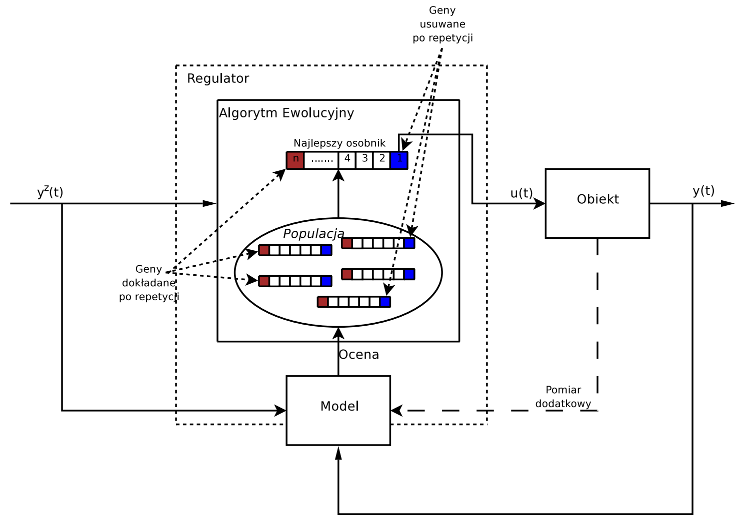
Układ regulacji predykcyjnej



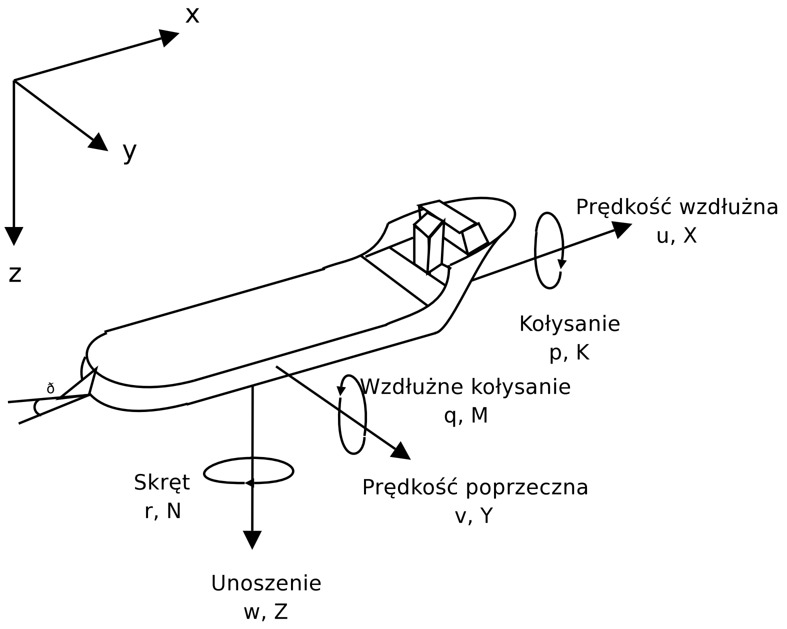
Układ regulacji predykcyjnej II



Algorytm ewolucyjny jako optymalizator



Zastosowania do sterowania statkiem

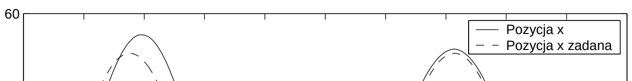


$$
\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & -mz_G & mx_G \\ 0 & -mz_G & I_{xx} & 0 \\ 0 & mx_G & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} =
$$

$$
\begin{bmatrix} X \\ Y \\ K \\ N \end{bmatrix} + \begin{bmatrix} m(vr + x_G r^2 - z_G pr) \\ -mur \\ mz_G ur \\ -mx_G ur \end{bmatrix}
$$

$$
\begin{aligned}
X' = {} & X'_{\dot{u}}\dot{u}' + X'(u') + (1-t)T' + X'_{vr}v'r' + X'_{rr}r'^2 + \\
& X'_v v' + X'_{vv}v'^2 + \\
& X'_{v\phi}v'\phi' + X'_{\phi}\phi' + X'_{\phi\phi}\phi'^2 + \\
& X'_{pp}p'^2 + X'_{ppu}p'^2 u'_a
\end{aligned}
$$

$$
\begin{aligned}
Y' = {} & Y'_{\dot{v}}\dot{v}' + Y'_{\dot{r}}\dot{r}' + Y'_{\dot{p}}\dot{p}' + \\
& Y'_v v' + Yvv'v'^2 + Y'_{v|v|}v'|v'| + Y'_{v|r|}v'|r'| + Y'_{vrr}v'r'^2 + \\
& Y'_r r' + Y'_{r|r|}r'|r'| + Y'_{rrr}r'^3 + Y'_{r|v|}r'|v'| + Y'_{rvv}rv'^2 + \\
& Y'_p p' + Y'_{ppp}p'^3 + Y'_{pu}p'u'_a + Y'_{pu|pu|}p'u'_a|p'u'_a| + \\
& Y'_{\phi}\phi' + Y'_{v\phi}v'\phi' + Y'_{v\phi\phi}v'\phi'^2 + Y'_{\phi vv}\phi'^2 v'^2 + Y'_0 + Y'_{0u}u'_a
\end{aligned}
$$



Problem przykładowy

Dany jest następujący problem; Znaleźć optymalne rozłożenie zadanej ilości elips na płaszczyźnie $\mathbb{R}^2$ tak, aby okrąg opisany na zbiorze tychże elips (czyli okrąg, którego promień jest wystarczająco duży, aby pomieścić wszystkie elipsy) miał jak najmniejszą średnicę. Parametry i ilość elips są danymi wejściowymi zadania. Dane jest k elips o równaniach

$$
\frac{(x - x_{k0})^2}{a^2} + \frac{(y - y_{k0})^2}{b^2} = 1
$$

gdzie a, b są danymi wejściowymi zadania. Algorytm powinien zwrócić

- Dobrany środek (x_{k0}, y_{k0}) dla każdej z elips,
- Środek i promień najmniejszego okręgu opisanego na tych elipsach