

ALGORYTMY EWOLUCYJNE W IDENTYFIKACJI ODWROTNYCH MODELI DIAGNOSTYCZNYCH

Grzegorz URBANEK

Katedra Podstaw Konstrukcji Maszyn, Politechnika Śląska
ul. Konarskiego 18 a, 44-100 Gliwice, fax: (32) 2371360 , email: gu@polsl.pl

Streszczenie

Jedną z metod identyfikacji modeli odwrotnych jest wyznaczanie ich na podstawie przykładów opisujących zadane stany rozpatrywanego obiektu oraz odpowiadające tym stanom symptomy. Stosowanie tej metody wymaga posiadania odpowiedniego zbioru takich przykładów, których jakość decyduje o jakości tak wyznaczanego modelu odwrotnego. W artykule opisano zastosowanie algorytmu ewolucyjnego do optymalizacji zbioru przykładów. Przedstawiono wyniki obliczeń, wskazano możliwości ograniczania czasu obliczeń.

Słowa kluczowe: modele odwrotne, algorytmy ewolucyjne, przybliżony symulator

EVOLUTIONARY ALGORITHMS IN IDENTIFICATION OF INVERSE DIAGNOSTICS MODELS

Summary

One of identification methods of inverse models is based on examples described determined states of examined object and symptoms corresponded to these states. Application of this method requires a proper set of such examples, the quality of which influence on quality of evaluated inverse model. The application of evolutionary algorithms for searching optimal set of examples is described in the paper. The results of the research and possibilities of reducing computation time are shown.

Keywords: inverse models, evolutionary algorithms, rough simulator

1. WSTĘP

Zastosowanie nowoczesnych komputerowych systemów nadzoru diagnostycznego wymaga opracowania skutecznych metod pozwalających na identyfikację relacji diagnostycznych, będących podstawą określania reguł tworzących bazy wiedzy systemów doradczych [3].

Jedną z takich metod jest pozyskiwanie modeli diagnostycznych poprzez odwracanie znanego modelu rozpatrywanego obiektu. Tak wyznaczony model nosi nazwę odwrotnego modelu diagnostycznego.

Odwracanie analityczne jest możliwe tylko dla prostych modeli matematycznych; powstaje problem jak odwrócić złożony model numeryczny?

W [3] zaproponowano prowadzenie identyfikacji modeli odwrotnych na podstawie przykładów opisujących zadane stany rozpatrywanego obiektu oraz odpowiadające tym stanom symptomy.

2. MODELE ODWROTNE

Modele odwrotne mogą być identyfikowane jako globalne lub lokalne. W [5] pokazano zastosowanie modeli globalnych identyfikowanych z użyciem sztucznych sieci neuronowych. Mimo uzyskania odpowiedniej jakości modelu odwrotnego

stwierdzono małą przydatność modeli globalnych ze względu na ich dużą złożoność (wyznaczanie i doskonalenie takich modeli jest niezwykle czasochłonne) oraz utrudnienia związane z próbą rozszerzenia uwzględnianych przykładów wzorcowych.

W [3] zaproponowano zastosowanie lokalnych modeli interpolacyjnych. Modele te są rozpięte na zbiorach przykładów zawartych w zbiorach danych wzorcowych (wymagają umieszczenia tego zbioru w definicji modelu; nie zastępują danych) co jest ich podstawową zaletą: nie przechowuje się modelu tylko zbiór danych, model wyznacza się w chwili zapotrzebowania na niego na podstawie aktualnych danych, a co za tym idzie rozszerzenie uwzględnianych przykładów wzorcowych nie stanowi problemu. Zwrócono uwagę, że bezpośredni wpływ na jakość wyznaczanego modelu odwrotnego ma jakość użytego zbioru przykładów.

W [7] przedstawiono pierwszą próbę doskonalenia zbioru przykładów wzorcowych; zaproponowany algorytm heurystyczny nie zawsze gwarantuje oczekiwane rozwiązanie.

W związku z wyraźnym podobieństwem zadania optymalizacji zbioru przykładów do zadań związanych z poszukiwaniem optymalnych populacji za pomocą algorytmów nazywanych algorytmami ewolucyjnymi, podjęto próbę

rozpatrywania tego zadania jako zadania poszukiwania optymalnej populacji przykładów.

3. ALGORYTMY EWOLUCYJNE

Algorytm ewolucyjny przetwarza populację osobników, z których każdy jest potencjalnym rozwiązaniem [1],[6],[8]. Każdy osobnik zawiera informację stanowiącą jego genotyp, który jest przepisem na utworzenie w wyniku oddziaływania środowiska fenotypu, czyli zbioru cech, na podstawie których funkcja oceniająca (funkcja przystosowania), określa przystosowanie osobnika. Celem algorytmu ewolucyjnego jest maksymalizacja funkcji przystosowania. Genotyp osobnika składa się z chromosomów (najczęściej jednego), które składają się z elementarnych części, zwanych genami. Istotą działania algorytmu ewolucyjnego przedstawia rys.1.

4. OPTIMALIZACJA ZBIORU PRZYKŁADÓW

Sposób optymalizacji zbioru przykładów, uzyskiwanych z badań symulacyjnych, przedstawiono na rys.2. Metoda ta jest rozwinięciem metody opisanej w [4].

4.1. Populacja

Populacja przetwarzana przez algorytm ewolucyjny składa się z N osobników. Genotypem osobnika jest zbiór X zawierający n przykładów $x = (s, v)$, gdzie: $x.s$ – wartości rozpatrywanych cech stanu (kodowane binarnie), $x.v$ – odpowiadające im wartości cech sygnałów diagnostycznych wyznaczone z zastosowaniem symulatora *Model*:

$$x.v = Model(x.s). \quad (1)$$

Fenotypem osobnika jest oszacowanie poszukiwanych wartości cech stanu $\hat{z}.s$, na podstawie genotypu X , dla zadanych wartości cech sygnałów diagnostycznych $z.v$, wyznaczone z

```

procedura Algorytm_ewolucyjny
{
  t:=0
  P[t]=inicjacja()
  ocena(P[t])
  dopóki (~warunek_zakończenia)
  {
    t:=t+1
    R[t]:=selekcja(P[t-1])
    D[t]:=krzyżowanie_i_mutacja(R[t])
    ocena(D[t])
    P[t]:=sukcesja(P[t-1],D[t])
  }
}

```

Rys. 1. Podstawowy algorytm ewolucyjny;
oznaczenia: t – czas (licznik kroków),
P – populacja, R – rodzice, D – dzieci

```

procedura Optymalizacja_AE
{
  t:=0
  P[t]:=inicjacja()
  ocena(P[t])
  {
    dla i:=1:N
    {
      oz.si:=invModel(X,z.v)
      oz.vi:=Model(oz.si)
      ocenai:=sim(oz.vi,z.v)
    }
  }
  dopóki (~warunek_zakończenia)
  {
    t:=t+1
    R[t]:=selekcja(P[t-1])
    D[t]:=krzyżowanie_i_mutacja(R[t])
    {
      ...
      jeżeli (jest_nowy x.sij)
      {
        x.vij:=Model(x.sij)
      }
    }
    ocena(D[t])
    {
      dla i:=1:N
      {
        oz.si:=invModel(X,z.v)
        oz.vi:=Model(oz.si)
        ocenai:=sim(oz.vi,z.v)
      }
    }
    P[t]:=sukcesja(P[t-1],D[t])
  }
}

```

Rys. 2. Optymalizacja zbioru przykładów z zastosowaniem algorytmu ewolucyjnego

użyciem algorytmu wielowymiarowej interpolacji *invModel*:

$$\hat{z}.s = invModel(X, z.v) \quad (2)$$

4.2. Funkcja przystosowania

Miarą stopnia przystosowania osobnika X jest wartość funkcji oceniającej $fit(X, z.v)$ wyznaczanej dla rozpatrywanego osobnika ze względu na zadane wartości cech sygnałów diagnostycznych $z.v$.

Osobnik X uznawany jest za maksymalnie przystosowany do określania modelu odwrotnego wtedy, gdy oszacowanie wartości cech sygnałów diagnostycznych $\hat{z}.v$ wyznaczone według (1) dla jego fenotypu $\hat{z}.s$ wyznaczonego według (2), jest zgodne z wartościami zadanymi $z.v$. Funkcja oceniająca fit może być definiowana jako miara podobieństwa *sim* pomiędzy wartościami oszacowanymi $\hat{z}.v$ a zadanymi $z.v$, przyjmująca wartości z przedziału od 0 do 1:

$$fit(X, z.v) = sim(\hat{z}.v, z.v) \in [0;1] \quad (3)$$

4.3. Selekcja i sukcesja

Przyjęto, że stosowana będzie selekcja proporcjonalna ([1],[6],[8]), natomiast sukcesja będzie prowadzona wg modelu elitarnego ([1],[6],[8]).

4.4. Krzyżowanie i mutacja

Dla założonego sposobu reprezentacji populacji, gdzie osobnikiem jest zbiór przykładów, opracowano odpowiedni sposób krzyżowania. Utworzenie zbioru potomnego ze zbiorów rodzicielskich przebiega następująco: do zbioru potomnego przepisuje się te przykłady, które występują w obu zbiorach rodzicielskich, brakujące (do zadanej liczby n) przykłady tworzy się poprzez „krzyżowanie właściwe” (dwupunktowe [1],[6],[8],[10]) przykładów losowo wybranych ze zbiorów rodzicielskich.

Przyjęto, że stosowana będzie mutacja nierównomierna ([1],[6],[8],[11]).

5. PRZYKŁAD ZASTOSOWANIA

5.1. Rozpatrywany układ

Opisaną metodę zastosowano do identyfikacji modeli odwrotnych układu przedstawionego na rys.3. Układ składa się z dwóch wałów dwupodporowych połączonych sprzęgłem.

Symulowaną niesprawnością było rozosiowanie wałów: jednoczesne przemieszczenie promieniowe (r) i kątowe (α). Przemieszczenia te były rozpatrywanymi cechami stanu.

Obserwowane były drgania podpór. Na podstawie analizy wrażliwości, jako symptomy symulowanej niesprawności, wybrano wartość szczytową dodatnią sygnału prędkości drgań obserwowanego na podporze nr 2 i 3.

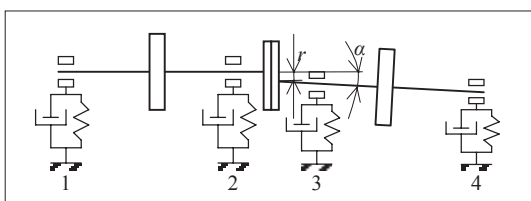
Badania były prowadzone przy stałej prędkości obrotowej wałów.

5.2. Parametry algorytmu

Dla potrzeb rozpatrywanego zadania przyjęto:

- populacja początkowa jest generowana losowo,
- liczba osobników populacji $N = 10$,
- liczba przykładów w zbiorze $n = 6$,
- stosowanym algorytmem wielowymiarowej interpolacji jest algorytm W13 ([3])
- prawdopodobieństwo krzyżowania = 1,
- prawdopodobieństwo mutacji = 0,1.

Warunki zakończenia działania algorytmu



Rys. 3. Rozpatrywany układ ([9])

ewolucyjnego:

- liczba pokoleń równa 100,
- przystosowanie najlepszego osobnika $\geq 0,99$.

6. SYMULATOR

Użycie symulatora (funkcja *Model*) ma miejsce gdy (rys.2):

- w wyniku operacji krzyżowania i mutacji powstaje nowy przykład, a właściwie zostają wygenerowane wartości cech stanu, dla których należy wyznaczyć wartości cech sygnałów diagnostycznych;
- istnieje potrzeba wyznaczenia wartości funkcji przystosowania dla osobnika populacji.

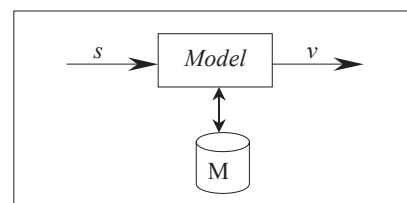
6.1. Symulator dokładny

W wyniku przeprowadzonych obliczeń (tab.1) okazało się, że zbiór przykładów o żądanej jakości (wartości funkcji przystosowania) otrzymywany był około 6 pokolenia, zaś symulator był użyty około 1000 razy.

W przypadku złożonych obiektów technicznych, których modele są zwykle złożonymi programami komputerowymi, koszt symulacji jest wysoki (ze względu na bardzo długi czas obliczeń). Ze względu na to stosowanie opisanej metody obliczeń dla takich obiektów byłoby niemożliwe (mimo uzyskiwania dobrych wyników).

6.2. Symulator przybliżony

Problemy związane ze stosowaniem symulatora dokładnego doprowadziły do powstania koncepcji symulatora przybliżonego (rys.4). Składa się on z symulatora dokładnego i pamięci wcześniejszych działań (zbioru wygenerowanych przykładów).



Rys. 4. Koncepcja symulatora przybliżonego; *Model* – symulator dokładny, M – pamięć przykładów

Symulator przybliżony ma wyznaczyć, zadaną dokładnością, wartości cech sygnałów diagnostycznych dla zadanych wartości cech stanu. Jego działanie składa się z dwóch kroków:

1. żądanie zadanej dokładności wartości cech sygnałów diagnostycznych przekształcane jest w żądanie znalezienia w pamięci przykładu, którego wartości cech stanu są w określonym otoczeniu zadanych wartości cech stanu; jeżeli taki przykład zostanie znaleziony, jest on przekazywany jako wynik działania symulatora (symulator dokładny nie został uruchomiony);
2. jeżeli próba znalezienia odpowiedniego przykładu w pamięci nie powiedzie się, uruchamiany jest

symulator dokładny, który wyznacza żądane wartości cech sygnałów diagnostycznych; tak wyznaczony nowy przykład jest wynikiem działania symulatora, trafia również do jego pamięci.

Założono, że wyznaczenie nowego przykładu w zbiorze będzie żądane ze średnią dokładnością, natomiast wyznaczenie cech sygnałów diagnostycznych dla oceny osobnika będzie żądane z wysoką dokładnością.

Przeprowadzone obliczenia dały następujące wyniki (tab.1):

1. pamięć symulatora była kasowana przed uruchomieniem obliczeń – rozwiązanie osiągnięte było ok. 8 pokolenia, symulator uruchamiany był ok. 100 razy;
2. pamięć symulatora nie była kasowana (zawarta w niej liczba przykładów była większa przy każdym uruchomieniu obliczeń) – rozwiązanie osiągnięte było w końcowych obliczeniach ok. 7 pokolenia, symulator uruchamiany był ok. 30 razy.

Zastosowanie symulatora przybliżonego doprowadziło do znacznego ograniczenia liczby uruchomień symulatora dokładnego, które dla złożonych obiektów stanowią główny koszt wyznaczenia modelu odwrotnego opisaną metodą.

Tab. 1. Otrzymane wyniki

wersja symulatora, liczba uruchomień algorytmu	początkowe uruchomienia algorytmu		końcowe uruchomienia algorytmu	
	rozwiązanie w pokoleniu (średnio)	liczba uruchomień symulatora (średnio)	rozwiązanie w pokoleniu (średnio)	liczba uruchomień symulatora (średnio)
dokładny, 50	6	1000	6	1000
przybliżony, 50 z kasowaniem pamięci	8	100	8	100
przybliżony, 10x50 bez kasowania pamięci	8	100	7	30

7. PODSUMOWANIE

Otrzymane wyniki potwierdzają przydatność stosowania algorytmu ewolucyjnego w optymalizacji przykładów do wyznaczania modeli odwrotnych, zastosowanie symulatora przybliżonego umożliwia zastosowanie tej metody dla złożonych obiektów rzeczywistych.

Niezależnie od przedstawionych sposobów optymalizacji zbioru przykładów można zastosować algorytm ewolucyjny do poszukiwania postaci funkcji interpolującej przykłady.

LITERATURA

- [1] Arabas J.: Wykłady z algorytmów ewolucyjnych, WNT, Warszawa 2001.

- [2] Cholewa W., Kiciński J.: Metody identyfikacji reguł dla diagnostycznych systemów doradczych. XXXV Sympozjon Modelowanie w Mechanice t.1, str. 57-66, Zeszyty Naukowe Katedry Mechaniki Technicznej, Gliwice, 1996.
- [3] Cholewa W., Kiciński J. (red.): Diagnostyka techniczna. Odwrotne modele diagnostyczne, Wydawnictwo Politechniki Śląskiej, Gliwice 1997.
- [4] Cholewa W., Urbanek G.: Optimization of examples for identification of inverse diagnostic models, Symposium on Methods of Artificial Intelligence in Mechanics and Mechanical Engineering AI-MECH 2001, Gliwice 2001.
- [5] Cholewa W., White M.F.: Application of neural networks for inverting of vibration models. Politechnika Śląska, Gliwice, 1994. Zeszyty naukowe – Mechanika z.116.
- [6] Goldberg D.E.: Algoritmy genetyczne i ich zastosowania, WNT, Warszawa 1998.
- [7] Klimek A.: Metody doskonalenia odwrotnych modeli diagnostycznych. Politechnika Śląska, Gliwice, 1999. Zeszyty naukowe – Mechanika z.134.
- [8] Michalewicz Z.: Algoritmy genetyczne + struktury danych = programy ewolucyjne, WNT, Warszawa 1999.
- [9] Sławik D.: Generator drgań. Raport KPKM.
- [10] Urbanek G.: Czynniki wpływające na jakość i koszt rozwiązania w identyfikacji modeli odwrotnych z zastosowaniem algorytmu ewolucyjnego. VI Krajowa Konferencja Naukowo-Techniczna Diagnostyka Procesów Przemysłowych DPP'03, Władysławowo 2003.
- [11] Urbanek G.: Factors having an effect on quality and cost of solution in identification of inverse models with the application of evolutionary algorithms, Symposium on Methods of Artificial Intelligence AI-METH 2003, Gliwice 2003.



Mgr inż. Grzegorz URBANEK jest absolwentem Wydziału Mechanicznego Technologicznego Politechniki Śląskiej w Gliwicach. W roku 1999 rozpoczął studia doktoranckie w Katedrze Podstaw Konstrukcji Maszyn tejże uczelni. Jego zainteresowania skupiają się głównie wokół meto-

dot komputerowego wspomaganie projektowania i eksploatacji maszyn, sztucznej inteligencji oraz diagnostyki technicznej. W roku 1999 przebywał na stypendium w Fachhochschule Schweinfurt-Würzburg-Aschaffenburg (Niemcy) w ramach programu Socrates/Erasmus