Obliczenia inspirowane Naturą

Wykład 07 - Genetyka i automaty (uzupełnienie wykładu 06)

Jarosław Miszczak

IITiS PAN Gliwice

21/04/2016

Wprowadzenie

Wprowadzenie

- Algorytmy genetyczne nie odnoszą się do konkretnego problemu – podają jedynie ogólny schemat (inspiracje) do tworzenia algorytmów.
- Wybór reprezentacji zależy od problemu.
- Wybór operatorów genetycznych (mutowania, krzyżowania) zależy od problemu.
- Czy dobór naturalny nadaje się do pisana programów?

Hodowla automatów komórkowych

Hodowla automatów komórkowych

Rozpoznawanie (własności) obrazów

- Problem: odróżnianie obrazu czarnego od białego – dla obrazu na sieci stwierdź, czy więcej je komórek białych czy czarnych.
- Wejściem jest ciąg jedynek (czarny) i zer (biały).
- Jeżeli automat doprowadzi do ciągu samych jedynek, to uznajemy, że automat rozpoznał ciąg wejściowy jako bardziej czarny. Jeżeli doprowadził do samych zer – że uznał wejście za "raczej białe".

M. Mitchell, J.P. Crutchfield, P.T. Hraber, *Evolving cellular automata to perform computations: mechanisms and impediments*, Physica D, 75(1), pp. 361-391 (1994)

Hodowla automatów komórkowych

Rozpoznawanie (własności) obrazów

Procedura oceniania automatów (wg. Mitchell, Crutchfielda i Hrabera) polegała na 100 eksperymentach według schematu:

- Wygeneruj losowy ciąg o zadanej gęstości jedynek (i długości 149).
- Potraktuj ten ciąg jako stan początkowy.
- Wykonaj 320 kroków.
- Jeżeli stan jest jednorodny, to przyznaj jeden punkt jeżeli automat dobrze ocenił gęstość.

Ocena automatu jest średnim wynikiem po wszystkich eksperymentach.

Hodowla automatów komórkowych

Rozpoznawanie (własności) obrazów

Rozważamy automaty jednowymiarowe $(2, 3)$, czyli mamy ... reguł.

- Zaczynamy od 100 losowych automatów.
- Oceniamy i zostawiamy 20 najlepszych.
- Wykonujemy krzyżowanie, żeby wygenerować kolejnych 80.
- Dodatkowo mutujemy – losowo zmieniamy dwa elementy w funkcji przejścia.

Hodowla automatów komórkowych

Uczenie automatów

Ewolucję automatów można podzielić na epoki.

- W pierwszej epoce najlepsze są automaty które dają zawsze ciąg stały.
- W drugiej epoce pojawiają się automaty klasyfikujące poprawnie skrajne przypadki.
- W trzeciej epoce pojawiają się automaty które stopniowo podejmują prawidłową decyzje dla coraz większego zakresu gęstości jedynek.

Szczytowe osiągniecie to 0.936.

Hodowla automatów komórkowych
Reguła GKL

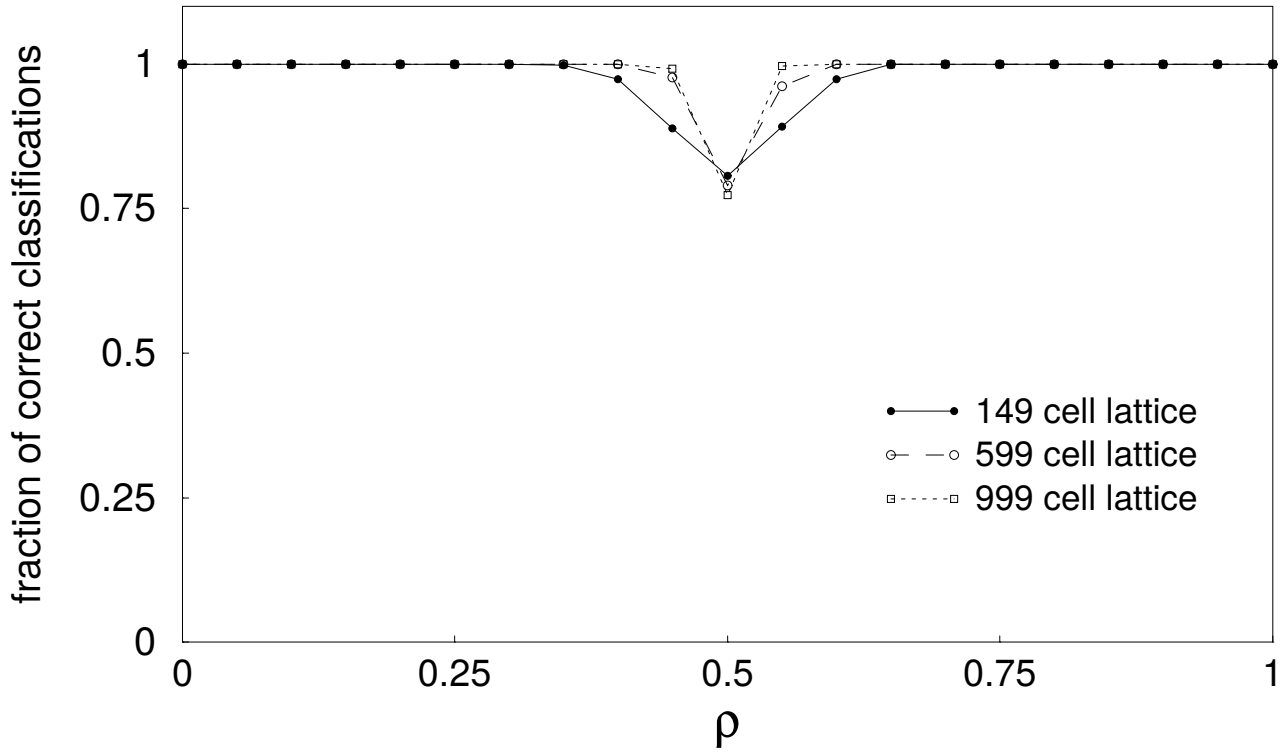
Reguła GKL (Gacsa-Kurdyumova-Levina)

- Jeżeli $s_i(t) = 0$, to $s_i(t+1)$ jest wybierany przez głosowanie ze zbioru $\{s_i(t), s_{i-1}(t), s_{i-3}(t)\}$
- Jeżeli $s_i(t) = 1$, to $s_i(t+1)$ jest wybierany przez głosowanie ze zbioru $\{s_i(t), s_{i+1}(t), s_{i+3}(t)\}$

P. Gach, G. L. Kurdyumov, L. A. Levin, *One-Dimensional Uniform Arrays That Wash Out Finite Islands*, Probl. Peredachi Inf., 14:3 (1978), 92–96

Hodowla automatów komórkowych
Reguła GKL

Reguła GKL (Gacsa-Kurdyumova-Levina) osiąga 0.972 – takiej wydajności nie osiągają automaty wyhodowane genetycznie.



Za: M. Mitchell, P. Hraber, J.P. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. Complex Systems, 7(2):89–130, 1993. arXiv:adap-org/9303003

Genetyczne pisanie programów

Genetyczne pisanie programów
Motywacja

- 1964, Lawrence J. Fogel – odkrywanie gramatyki języka dla którego znany jest alfabet i przykłady. Ewolucji poddawane były automaty skończone.
- Automatyczne generowanie programów w języku LISP.
- 1981, Richard Forsyth – wykorzystanie struktury drzewiastej do klasyfikacji chorób serca.

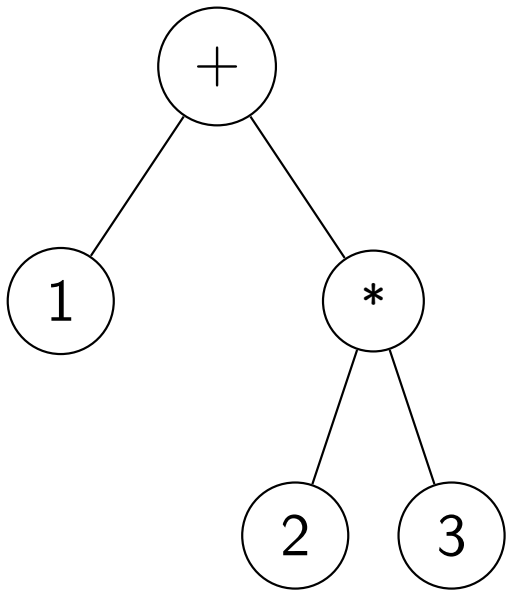
Genetyczne pisanie programów
Reprezentacja programów

- Chromosom jest kodowany jako drzewo (węzły i krawędzie).
- Najlepiej odpowiada to językom funkcyjnym – oryginalnie LISP.
- Abstrakcyjne Drzewo Składni daje podobną do LISPa strukturę.
- Garnett Wilson, Wolfgang Banzhaf, 1998 – liniowe programowanie genetyczne, przystosowane do języków imperatywnych.
- μGP (MicroGP) (http://ugp3.sourceforge.net/) – optymalizacja kodu asmeblerowego.

Genetyczne pisanie programów

Reprezentacja programów

W języku LISP program $(+\ 1\ (*\ 2\ 3\ ))$ można zapisać jako drzewo.

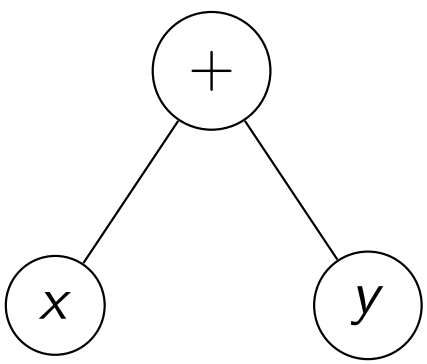


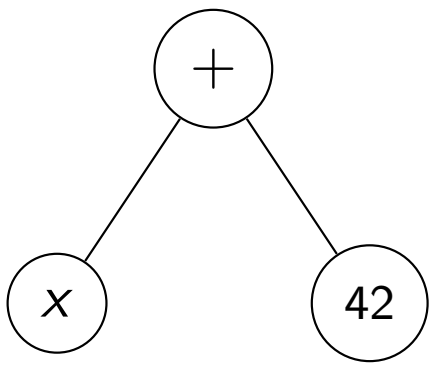
Węzłami są operatory, a liśćmi – operandy.

Taka forma reprezentacji programu jest nazywana abstrakcyjnym drzewem składni (ang. *Abstract Syntax Tree*, AST).

Genetyczne pisanie programów
Możliwe optymalizacje

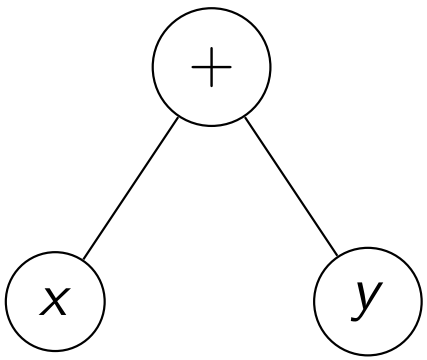
Najprostsza mutacja to zmiana węzła terminalnego.

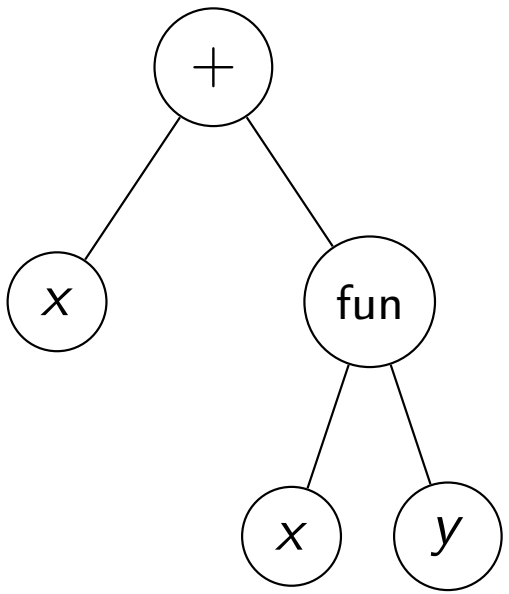




Genetyczne pisanie programów
Możliwe optymalizacje

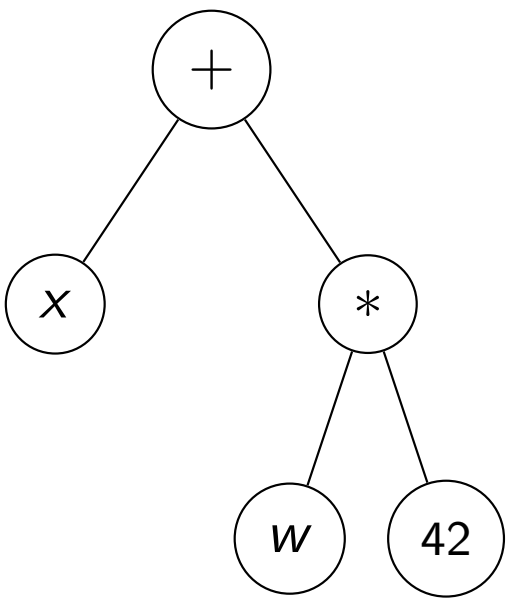
Możemy też dołączyć losowe poddrzewo.

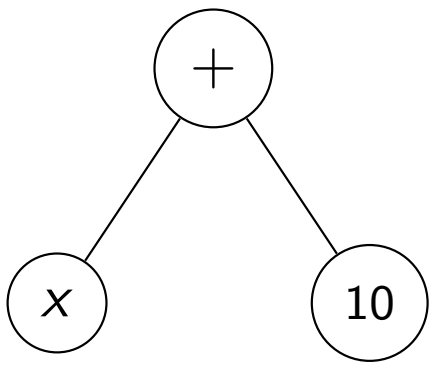




Genetyczne pisanie programów
Możliwe optymalizacje

Albo zamienić poddrzewo na węzeł teminalny.





Genetyczne pisanie programów
Funkcja celu

- Funkcja celu powinna określać jak dobrze program rozwiązuje zadany problem.
- W przypadku automatów skończonych do oceny może służyć odsetek poprawnie rozpoznanych przykładów.

Genetyczne pisanie programów
Ciekawe zastosowania

- Optymalizacja algorytmów kwantowych (L. Spector, H. Barnum, H.J. Bernstein, N. Swamy, Quantum Computing Applications of Genetic Programming, Advances in genetic programming, pp. 135-160 (1999).)
- Synteza mechanizmów optymalnej m.in. kontroli robotów, trasowania i reakcji chemicznych (J.R. Koza, M.J. Streeter, M.A. Keane, *Routine high-return human-competitive automated problem-solving by means of genetic programming*, Information Sciences, 178(23), pp. 443-4452 (2008)).

Genetyczne pisanie programów
Ciekawe zastosowania

http://genetic-programming.org/