

## 18. Podstawowe metody „obliczeń miękkich (inteligentnych)”.

### 30. Jakie techniki wchodzą w skład obliczeń miękkich. Charakterystyka każdej z nich.

#### [Wstęp]

Wiele problemów w optymalizacji nie ma dokładnego algorytmów rozwiązania, albo ten algorytm ma tak dużą złożoność obliczeniową, że w praktyce i tak go nie ma. Wtedy stosuje się heurystyki, które nie ma gwarancji znalezienia rozwiązania optymalnego. Nie ma nawet gwarancji uzyskania rozwiązania proporcjonalnego do optymalnego, jak w algorytmach przybliżonych (aproksymacyjnych).

Metody używa się też często do znajdowania rozwiązań przybliżonych, na podstawie, których później wylicza się ostateczny rezultat pełnym algorytmem. To ostatnie zastosowanie szczególnie dotyczy przypadków, gdy heurystyka jest wykorzystywana do nakierowywania pełnego algorytmu ku optymalnemu rozwiązaniu, aby zmniejszyć czas działania programu w typowym przypadku bez poświęcania jakości rozwiązania (np. algorytm  $A^*$  w odnajdywaniu najkrótszej ścieżki w grafie).

#### [Odpowiedz]

**Sztuczne systemy immunologiczne** – stanowią odpowiednik realizacji procesu adaptacji i dywersyfikacji naturalnego systemu immunologicznego. Ich zadaniem jest, poprzez sterowanie populacją przeciwciał, doprowadzenie do otrzymania rozwiązania. Algorytmy immunologiczne można podzielić na populacyjne (selekcja klonalna, selekcja negatywna) i sieciowe (sieć idiotypowa).

**Techniki rojowe** (mrówkowe, pszczele, świetlikowe, kukułcze) - wywodzące się z algorytmu optymalizacji kolonii cząstek - particle swarm optimization PSO. W zależności od stworzenia, na którego obserwacjach bazuje metoda, przeszukiwanie dziedziny i wybór najlepszych rozwiązań następuje w różny sposób, wszystkie mają jednak jedną bazę i w zapisie formalnym różnią się nieznacznie.

**Oparte o teorię chaosu** - bazujące na deterministycznych układach, tak wrażliwych na drobne zmiany warunków początkowych, że podczas obserwacji uchodzą za działające losowo (niedeterministycznie)

**Logika rozmyta typu 2** - wprowadzająca stany pośrednie w logice, umożliwiające określenie stopnia przynależności obiektu do zbioru.

Dzieli się na przedziałowe i uogólnione:

- przedziałowe: współczynniki przynależności są przedziałami ostrymi, umożliwiają modelowanie niepewności w przeciwieństwie do logiki rozmytej typu 1, a operacje and i or są proste i szybkie,
- uogólnione: współczynniki przynależności nie są ostre, należą do zbioru rozmytego. Podejście na razie raczej teoretyczne ze względu na słabo poznane zasady matematyczne i złożoność obliczeniową (zamiast operacji and i or - meet i join, bardzo wymagające obliczeniowo).

Przykładowo:

- typ-1: Karol jest w 0.72 wysoki,

- typ-2 przedziałowy: Karol jest w  $[0.62-0.82]$  wysoki, rozkład przynależności równomierny

- typ-2 uogólniony: Karol jest w  $N(0.72, 0.1)$  wysoki, rozkład przynależności normalny (mam nadzieję że nie zakręciłem) czy jakkolwiek inny niż równomierny

**Techniki agentowe** - system oparty o autonomiczne byty zwane agentami, rozproszone podejście do rozwiązywania problemu.

Agent:

- oddziałuje na środowisko,
- komunikuje się z innymi agentami,
- działa realizując wyznaczone cele,
- ma dostęp i dysponuje zasobami,
- posiada jakiś zbiór umiejętności,
- posiada ograniczoną percepcję,
- posiada wiedzę nt środowiska lub ją gromadzi
- czasami może się rozmnażać (pewnie jak spotka ładną agentkę :) )

**Zbiory przybliżone** - rough sets, dla zbiorów o nieregularnych zakresach definiujemy przybliżenie górne i dolne o zakresach regularnych. Dzięki temu możemy określić nieostre pojęcie w ścisły sposób. Przynależność sprawdza się na podstawie klas równoważności  $R$ , zwanych atomami. Obiekty należące do tej samej klasy równoważności są nierozróżnialne.

- aproksymacja dolna: składa się z obiektów, które z całkowitą pewnością należą do zbioru  $X$
- aproksymacja górna: składa się z obiektów, które MOGĄ należeć do zbioru  $X$
- obszar brzegowy: różnica między aproksymacją dolną i górną
- zbiór dokładny (crisp): obszar brzegowy nie zawiera żadnych obiektów
- zbiór przybliżony (rough): obszar brzegowy zawiera jakieś obiekty

**Hybrydy** - rozwiązania powstałe poprzez połączenie powyższych z innymi rozwiązaniami, np. rozmyte sieci neuronowe, transformata falkowa połączona z sieciami lub też połączenie między sobą itd.

## Algorytm ewolucyjny

Algorytm genetyczny jest algorytmem przeszukującym przestrzeń rozwiązań. Jest podobny do algorytmów iteracyjnych i losowych, z tą różnicą, że operuje na zbiorze rozwiązań, a nie na poszczególnym jednym rozwiązaniu, przeprowadzając na nim równoległe obliczenia.

Inspiracją algorytmów genetycznych dla J.H.Hollanda był proces ewolucji w przyrodzie. Przez to połączenie stosuje on specyficzną nomenklaturę jak na algorytm optymalizacji.

Optymalizacja	Algorytm genetyczny
Zmienna optymalizacyjna	Osobnik (chromosom)
Zbiór rozwiązań dopuszczalnych	Populacja
Funkcja celu	Funkcja przystosowania
Iteracja	Pokolenie
Kroki (fazy) algorytmu optymalizacji	Operatory

Do zastosowania algorytmu genetycznego do konkretnego problemu, należy określić pięć rzeczy: reprezentację rozwiązania, funkcję przystosowania oraz operatory selekcji, krzyżowania i mutacji.

Operatory selekcji wybierają, które osobniki przechodzą do nowej populacji. Powinno to zależeć od funkcji celu, ergo jeśli  $f(A) > f(B)$ , to osobnik B powinien mieć większy udział w nowej populacji niż A. Operatory krzyżowania biorą dwóch osobników i zwracają ich modyfikację. Uzyskane rozwiązania powinny mieć cechy obu wejściowych. Operatory mutacji modyfikują jedno rozwiązanie.

Schemat algorytmu genetycznego jest następujący:

1. Utworzenie początkowego zbioru rozwiązań  $P_0$ ,  $l := 0$

2. Jeżeli nie warunek stopu:

1. Selekcja: Przy pomocy operatorów selekcji wybierz osobniki z  $P_{l-1}$  do nowej populacji  $P_l$ .
2. Krzyżowanie: Zmodyfikuj losowe pary osobników z  $P_l$  przy pomocy operatorów krzyżowania.
3. Mutacja: Zmodyfikuj losowe osobniki z  $P_l$  przy pomocy operatorów mutacji.
4.  $l := l + 1$

## Symulowane wyżarzanie

Tło fabularne: Metoda ta bazuje na pewnej analogii termodynamicznej a w szczególności ze sposobem, w jaki ciecz zamarza i krystalizuje lub w jaki metale stygną i wyżarzają się. W wysokich temperaturach molekuly cieczy poruszają się swobodnie względem siebie. W wyniku schładzania ruchy termiczne ulegają zmniejszeniu. Jeżeli schładzanie jest powolne, to molekuly zwykle tworzą uporządkowane struktury - kryształy. Kryształ jest stanem o minimum energii tego systemu. Jeżeli natomiast ciecz jest chłodzona szybko to zwykle nie osiąga stanu krystalicznego, ale pozostaje w pewnym stanie amorficznym (szklistym) posiadającym wyższą energię.

W celu wytłumaczenia takiego zachowania się układów fizycznych musimy odwołać się do rozkładu Boltzmann'a:  $P(E) = \exp\left(-\frac{E_j - E_i}{kT}\right)$ , który określa prawdopodobieństwo znalezienia się układu utrzymywanego w temperaturze  $T$  w stanie o energii  $E_j$ , jeśli wcześniej był w stanie o energii  $E_i$  ( $k$  jest tzw. stałą Boltzmann'a). Z wzoru tego wynika, że zawsze istnieje możliwość znalezienia się danego układu w stanie o wyższej energii.

**Algorytm:** Ideą algorytmu Metropolis'a (z tła fabularnego) jest to, że algorytm dopuszcza czasami gorsze rozwiązania, w nadziei, że z nich trafi do lepszych. Symulowane wyżarzanie modyfikuje ten algorytm o zmienną temperatury  $T$ , która powinna maleć w czasie działania algorytmu. Więc na początku działania algorytm będzie częściej akceptował gorsze rozwiązania niż na końcu. Algorytm zmiany temperatur  $T$  nazywany jest schematem schładzania i może być zwykłą funkcją liniową, jak i skomplikowanym potworkiem.

Do zastosowania algorytmu symulowane wyżarzanie do konkretnego problemu, należy określić: reprezentację rozwiązania, funkcję przystosowania, schemat schładzania oraz generowanie nowych stanów do których algorytm może przeskoczyć.

1.  $s \leftarrow s_0, s_{best} \leftarrow s$
2.  $e \leftarrow E(s), e_{best} \leftarrow e$
3.  $l \leftarrow 1$
4. while  $l < L$ 
  - 4.1.  $T \leftarrow \text{cooling\_scheme}\left(\frac{l}{L}\right)$
  - 4.2.  $s_{new} \leftarrow \text{neighbour\_generator}(s)$
  - 4.3.  $e_{new} \leftarrow E(s_{new})$
  - 4.4. if  $\exp\left[-\frac{s_{new} - s}{T}\right] > \text{random}$  then
    - 4.4.1.  $s \leftarrow s_{new}$
    - 4.4.2.  $e \leftarrow e_{new}$
  - 4.5. if  $e < e_{best}$  then
    - 4.5.1.  $s_{best} \leftarrow s$
    - 4.5.2.  $e_{best} \leftarrow e$
  - 4.6.  $l \leftarrow l + 1$

return  $s_{best}$

## Tabu search

Tabu search (TS) jest algorytmem do optymalizacji dyskretnej. Przeszukuje on wszystkie rozwiązania, sąsiadujące do tego, w którym się znajduje. Pomysł polega na tym, by zabronić

powrotu do już odwiedzonych rozwiązań (na co najmniej kilka iteracji), czyli algorytm może wybrać gorsze rozwiązanie, by podążać inną ścieżką. Możliwe jest zastosowanie tego algorytmu, także do przestrzeni ciągłych poprzez ich dyskretyzację.

Do zastosowania TS do konkretnego problemu, należy określić: reprezentację rozwiązania, funkcję przystosowania oraz generowanie sąsiedztwa dla rozwiązania.