

Obliczenia z wykorzystaniem sztucznej inteligencji

wykład I

Wprowadzenie do inteligencji obliczeniowej

Optymalizacja

Joanna Kołodziejczyk

2020

Plan wykładu

- 1 Inteligencja obliczeniowa
 - Definicja i metody w CI
 - Zalety i wady CI
- 2 Optymalizacja
- 3 Hill climbing
- 4 Symulowane wyżarzanie
- 5 Literatura

Co to jest inteligencja obliczeniowa?

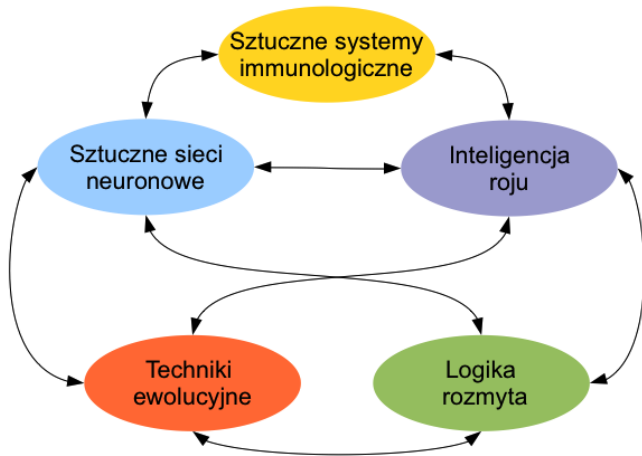
Inteligencja obliczeniowa — Computational intelligence

Do tej dziedziny zalicza się wszelkie podejścia i technologie do analizy, projektowania i tworzenia systemów inteligentnych. Tak szerokie zastosowanie pojęcia CI zostało sformalizowane przez IEEE Neural Network Council and IEEE World Congress on Computational Intelligence w Orlando latem 1994 roku.

Głównie dotyczy połączenia (wykorzystania) systemów takich jak:

- sztuczne sieci neuronowe
- systemy rozmyte
- obliczenia ewolucyjne
- sztuczne systemy immunologiczne
- inteligencja stadna (roju)
- inne technologie związane z budową inteligentnych agentów.

Możliwości łączenia pięciu głównych technik CI



Możliwości inteligencji obliczeniowej

- Duże zainteresowanie dziedziną doprowadziło do znaczących odkryć teoretycznych i technologicznych.
- Dziedzina uzyskała uznanie i akceptację w środowisku naukowym i wśród użytkowników.
- Techniki stosowano z dużą skutecznością do rozwiązywania licznych i rzeczywistych problemów.
- CI wskazało nowe obszary dla badań naukowych jak i zastosowań w przemyśle i biznesie.
- Osiągnięcia w tej dziedzinie rodzą zainteresowanie i są inspiracją dla praktyków z przemysłu i branż usługowych.
- Dziedzina jest silnie interdyscyplinarna, co skutkuje przełomowymi odkryciami i przełamywaniem barier technologicznych.

Sztuczne Sieci neuronowe

- Inspirowane naturalnym układem nerwowym
- Możliwości uczenia się, zapamiętywania i generalizacji
- Techniki
 - Perceptron
 - MLP, RBF
 - Hopfield
 - Kohonen
 - sieci jednokierunkowe i rekurencyjne głębokie
- Zastosowania
 - Funkcje, szeregi czasowe - aproksymacja
 - Sterowanie i optymalizacja
 - Rozpoznawanie wzorców / klasyfikacja
 - Klasteryzacja
 - Pamięć asocjacyjna

Obliczenia ewolucyjne

- Naśladują procesy przystosowawcze opisane w koncepcji Darwinowskiej
- Możliwości optymalizacji
- Techniki
 - Algorytmy genetyczne
 - Programowanie genetyczne
 - Strategie ewolucyjne
 - Ewolucja różnicowa
- Zastosowania
 - Drążenie danych
 - Optymalizacja kombinatoryczna
 - Detekcja błędów
 - Klasyfikacja
 - Aproksymacja w szeregach czasowych
 - Projektowanie

Inteligencja roju

- Naśladują procesy zachodzące w naturze najczęściej w organizmach zbiorowych
- Możliwości optymalizacji
- Techniki
 - Algorymy mrówkowe
 - PSO
 - ABC
 - wiele innych
- Zastosowania
 - Znajdowanie najkrótszych ścieżek w grafie
 - Kolorowanie grafu
 - Harmonogramowanie
 - Grupowanie

Sztuczne systemy immunologiczne

- Naśladują procesy zachodzące w naturalnym systemie immunologicznym
- Możliwości optymalizacji i wykrywania anomalii poprzez odróżnianie patogenów
- Techniki
 - Selekcja klonalna
 - Sieci idiotypowe
- Zastosowania
 - Rozpoznawanie wzorców
 - Klasyfikacja
 - Grupowanie

Systemy rozmyte

- Zainspirowane ludzkim rozumowaniem
- Wioskowanie w niepewności
- Techniki
 - Fuzzy Inference Systems *Mamdani, *Takagi-SUgeno-Kang
 - Fuzzy C-Means
- Zastosowania
 - Systemy sterowania
 - Klasyfikacja
 - Grupowanie

Plan wykładu

1 Inteligencja obliczeniowa

2 Optymalizacja

- Zadania optymalizacji i sposoby ich rozwiązywania
- Taksonomia metod optymalizacji
- Typy problemów optymalizacyjnych
- Jedno i wielokryterialne funkcje celu
- Dodatkowa frazeologia

3 Hill climbing

4 Symulowane wyżarzanie

5 Literatura

Optymalizacja

Poszukiwanie stanu optymalnego jest jednym z fundamentalnych zadań. Rozwiązują je cząstki, organizmy, oraz my na różne potrzeby. Zadanie optymalizacji może być sformalizowane:

Optymalizacja globalna

Jest to dziedzina matematyki i analizy numerycznej, która dotyczy optymalizacji. Celem optymalizacji jest znalezienie takiego elementu x^* , ze zbioru elementów \mathbb{X} , który spełnia zbiór kryteriów optymalizacji $F = \{f_1, f_2, \dots, f_n\}$. Kryteria wyrażone są w postaci funkcji.

Funkcja celu

Funkcja celu (objective function)

Jest to odwzorowanie $f : \mathbb{X} \rightarrow Y$, gdzie $Y \in \mathbb{R}$, które jest przedmiotem optymalizacji.

- Przeciwdziedzina funkcji celu jest podzbiorem liczb rzeczywistych.
- Dziedzina jest nazywana „przestrzenią problemu” i może być reprezentowana przez dowolne elementy numeryczne, listy i każdą dowolną strukturę danych.
- Rozwiązanie zadania optymalizacyjnego wymaga wybrania reprezentacji problemu.
- Funkcja celu niekoniecznie musi być czystym wyrażeniem matematycznym, może być złożonym algorytmem, który np. wymaga szeregu symulacji.

Heurystyki

Heurystyka

Jest częścią algorytmów optymalizacji, która używa informacji zebranych dotychczas przez algorytm lub danych z góry, wspomagających podjęcie decyzji, który z kandydatów na rozwiązanie powinien być testowany i jak utworzyć następnego kandydata na rozwiązanie. Heurystyka jest zależna od rozwiązywanego problemu.

Funkcja heurystyczna

Funkcja odwzorowująca stany (kandydatów na rozwiązanie) we współczynnik ich użyteczności

$$h : \mathbb{X} \rightarrow \mathbb{R}$$

gdzie \mathbb{X} przestrzeń stanów, a \mathbb{R} zbiór liczb rzeczywistych.

Metaheurystyki

Metaheurystyka

Jest metodą rozwiązywania dużej klasy problemów. Łączy funkcję celu lub heurystykę w abstrakcyjny i skuteczny sposób, zazwyczaj bez głębszego spojrzenia na specyfikę rozwiązywanego problemu. Traktuje się rozwiązywany problem jako black-box.

Przykładem może być symulowane wyżarzanie, które wybiera kandydata na rozwiązanie według prawdopodobieństwa wyznaczanego na podstawie współczynnika Boltzmana studzenia atomów metalu.

Podział według czasu optymalizacji

On-line

Zadanie musi zostać rozwiązane szybko (w czasie rzeczywistym). Kosztem szybkiego znalezienia rozwiązania jest jego jakość (optymalność). Np. roboty.

Off-line

Czas nie gra roli (na rozwiązanie można czekać nawet kilka dni), za to wyniki są bliskie optimum. Np. eksploracja danych, harmonogramowania.

Problemy kombinatoryczne

To problemy, które są definiowane w skończonej (lub liczbowo nieskończonej) dyskretnej przestrzeni problemowej X i których kandydat na rozwiązanie może być wyrażany jako:

- elementy ze zbiorów skończonych
- skończona sekwencja lub permutacja elementów x_i wybranych ze skończonych zbiorów $x \in X \rightarrow x = (x_1, x_2, \dots)$
- zbiór elementów $x \in X \rightarrow x = \{x_1, x_2, \dots\}$
- struktury drzew lub grafów o właściwościach węzłowych lub brzegowych wynikających z jednego z powyższych typów
- dowolnej formy mieszanej z powyższych.

Przykłady problemów kombinatorycznych

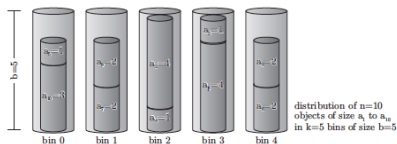
- problem komiwojażera, Traveling Salesman Problem
- Vehicle Routing Problems
- graph coloring
- graph partitioning
- scheduling
- packing
- satisfiability problems

Bin Packing Problem

Podane są liczby

- $k \in \mathbb{N}$ pojemników
- z których każdy ma wielkość $b \in \mathbb{N}$,
- $n \in \mathbb{N}$ elementów o wagach a_1, a_2, \dots, a_n .

Cel: Czy n obiektów może być rozłożone w k pojemnikach w taki sposób, aby żaden z nich nie został przepełniony?



Bin Packing Problem

Problem można sformalizować pytając, czy istnieje odwzorowanie x z liczb $1..n$ do $1..k$, tak aby nierówność

$$\exists x : \forall i \in 0..k - 1 : \left(\sum_{\forall j \in x[i]} a_j \right) \leq b$$

było spełniona. Wynikiem jest lista x k zbiorów, gdzie każdy i -ty zbiór zawiera indeksy obiektów zapakowanych do i -tego pojemnika.

Lub znaleźć odwzorowanie, gdzie funkcja, przyjmuje najmniejszą wartość:

$$f_{bp}(x) = \sum_{i=0}^{k-1} \max \left\{ 0, \left(\sum_{\forall j \in x[i]} a_j \right) - b \right\}$$

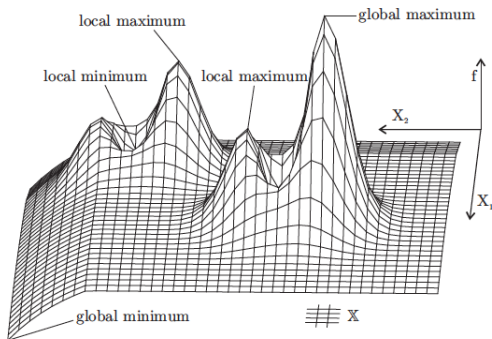
Problemy optymalizacji numerycznej

To problemy, które są definiowane w nieskończonej przestrzeni numerycznej takiej liczby rzeczywiste, czy złożone $\mathbb{X} \subseteq \mathbb{R}^n$. Problemy numeryczne zdefiniowane nad \mathbb{R}^n (lub dowolnym rodzajem przestrzeni zawierającej ją) nazywane są problemami ciągłej optymalizacji.

- Optymalizacja funkcji
- zadania optymalizacji projektów inżynierskich
- zadania klasyfikacji i eksploracji danych

Jednokryterialna funkcja celu

- Optymalizuje się jedną funkcję f zamiast ich zbioru.
- Poszukuje się albo jej maksimum, albo jej minimum.
- Jeżeli łatwo skonstruować f do maksymalizacji, a wykonać chcemy jej minimalizację, to przekształcamy cel na $(-f)$ lub $\frac{1}{f}$.



Optima

Lokalne maksimum

Lokalne maksimum \hat{x}_l funkcji celu $f : \mathbb{X} \rightarrow \mathbb{R}$ jest to taki element, że $f(\hat{x}_l) \geq f(x)$ dla wszystkich x sąsiadujących z \hat{x}_l .

Globalne maksimum

Globalne maksimum \hat{x} funkcji celu $f : \mathbb{X} \rightarrow \mathbb{R}$ jest to taki element, że $f(\hat{x}) \geq f(x)$ dla wszystkich $x \in \mathbb{R}$.

Lokalne maksimum zatem jest maksimum w pewnym obszarze dziedziny, a nie w całej dziedzinie tak jak optimum globalne.

Funkcja może mieć więcej niż jedno optimum globalne (sin). Rozwiązaniem jest wówczas zbiór X^* .

Wielokryterialna funkcja celu

W rzeczywistych problemach najczęściej należy optymalizować zbiór funkcji celu F , z których każda f_i stanowi niezależne kryterium.

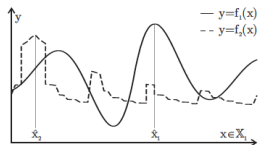
$$F = \{f_i : \mathbb{X} \rightarrow Y_i : 0 < i < n, Y_i \subseteq \mathbb{R}\}$$

Przykład zadania optymalizacji dla fabryki. Kryteria optymalizacji:

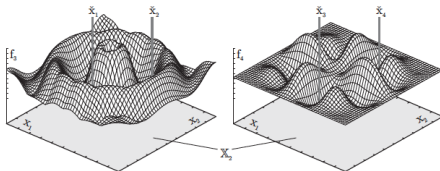
- Minimalizacja czasu pomiędzy zamówieniem a dostarczeniem gotowego produktu.
- Maksymalizacja zysku.
- Minimalizacja kosztów produkcji (zatrudnienia, materiałów, reklamy).
- Maksymalizacja jakości produktu
- Minimalizacja ilości odpadów.

Cel optymalizacji wielokryterialnej

Nie można mówić o globalnym optimum. Rozwiązaniem jest zbiór elementów optymalnych $x^* \in X^* \subseteq \mathbb{X}$. Ponieważ różne kryteria mogą być sprzeczne należy ustalić sposób kompromisu. Różne podejścia do jego ustalenia prowadzą do uzyskania różnego zbioru x^* .



różne maksima $\hat{x}_1 \neq \hat{x}_2$



różne minima $(\check{x}_1 = \check{x}_2) \neq (\check{x}_3 = \check{x}_4)$

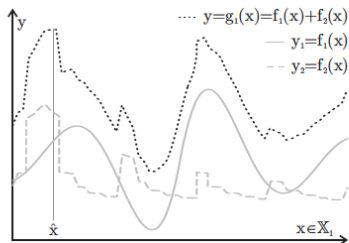
Kompromis 1— suma ważona

- Agregacja liniowa jest najprostszym sposobem określenia rozwiązania optymalnego.
- Każda funkcja (kryterium) f_i ma nadaną wagę w_i określającą jej istotność.
- Stosowanie znaku pozwala zamieniać cel kryterium z minimum na maksimum i odwrotnie.
- Redukuje się problem wielokryterialny do jednokryterialnego.

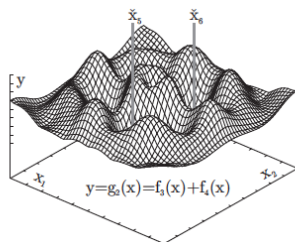
$$g(x) = \sum_{i=1}^n w_i f_i(x) = \sum_{\forall f_i \in F} w_i f_i(x)$$

$$x^* \in X^* \Leftrightarrow g(x^*) \geq g(x) \quad \forall x \in X$$

Kompromis 1— suma ważona



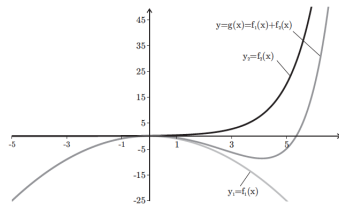
ustalone wagi $w_1 = w_2 = 1$



ustalone wagi $w_3 = w_4 = 1$

Wady podejścia ważonego

- Suma ważona jest dobrym rozwiązaniem, gdy wszystkie funkcje składowe sumy mają podobną złożoność (należą do tej samej grupy



złożoności w notacji wielkiego O).

- Osobnym problemem staje się ustalenie dobrego zestawu wag (kolejny problem optymalizacji).

Kompromis 2— optimum Pareto

- Optymalność w sensie Pareto z sukcesem była stosowana w ekonomii, teorii gier, naukach inżynierskich czy społecznych.
- W wyniku otrzymuje się granicę obejmującą wszystkie konfliktujące optima. Z tej granicy człowiek lub algorytm wybiera konfigurację, która jest najbardziej odpowiednia.

Dominacja

Element x_2 jest zdominowany przez element x_1 ($x_1 \vdash x_2$), jeżeli obiekt x_1 jest lepszy od x_2 biorąc pod uwagę jedną funkcję celu i nie jest gorszy biorąc pod uwagę inne funkcje celu.

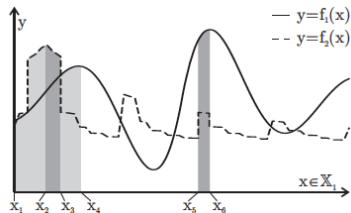
$$\begin{aligned}
 x_1 \vdash x_2 \Leftrightarrow & \forall i : 0 < i \leq n \Rightarrow \omega_i f_i(x_1) \leq \omega_i f_i(x_2) \wedge \\
 & \exists j : 0 < j \leq n : \omega_j f_j(x_1) < \omega_j f_j(x_2) \\
 \omega_i = & \begin{cases} 1, & \text{jeżeli } f \text{ jest minimalizowana} \\ -1, & \text{jeżeli } f \text{ jest maksymalizowana} \end{cases}
 \end{aligned}$$

Kompromis 2— optimum Pareto

Optimum w sensie Pareto

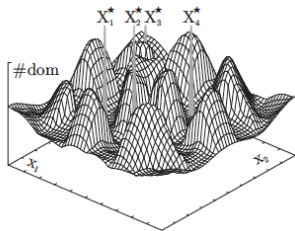
Element $x^* \in \mathbb{X}$ jest Pareto-optymalny jeżeli nie jest zdominowany przez inny element z przestrzeni problemu \mathbb{X} . X^* jest nazywane zbiorem Pareto (Pareto set) lub granicą Pareto (Pareto frontier).

$$x^* \in X^* \Leftrightarrow \nexists x \in \mathbb{X} : x \vdash x^*$$



$$\omega_1 = \omega_2 = -1$$

$$X^* = [x_2, x_3] \cup [x_5, x_6]$$



$$\omega_2 = \omega_3 = 1$$

$$X_1^*, X_2^*, X_3^*, X_4^*$$

Kilka definicji

Przestrzeń problemu (problem space)

Przestrzeń problemu \mathbb{X} problemu optymalizacyjnego jest to zbiór zawierający wszystkie elementy x , które mogą być rozwiązaniem problemu.

Kandydat na rozwiązanie (solution candidate)

Kandydat na rozwiązanie x jest elementem przestrzeni problemu \mathbb{X} pewnego problemu optymalizacyjnego.

Przestrzeń przeszukiwania (search space)

Przestrzeń przeszukiwania \mathbb{G} problemu optymalizacyjnego jest to zbiór wszystkich elementów g , które zostaną przeszukane w procesie poszukiwania rozwiązania.

Plan wykładu

- 1 Inteligencja obliczeniowa
- 2 Optymalizacja
- 3 Hill climbing**
 - Opis algorytmu
 - Wady i zalety
- 4 Symulowane wyżarzanie
- 5 Literatura

Cechy stochastycznych metod optymalizacji dla zadań dyskretnych

Stochastyczne metody optymalizacji operują na pojedynczym kandydacie na rozwiązanie z pewnej przestrzeni problemu i generują krok do kandydata sąsiedniego. Algorytmy nie są systematyczne (deterministyczne), ale mają następujące zalety:

- Małe wymagania pamięciowe: zazwyczaj stała wielkość.
- Często znajdują akceptowalne rozwiązania w dużej i nieskończonej przestrzeni rozwiązań, gdy metody deterministyczne zawodzą.

Metoda największego wzrostu — Hill climbing

- Metoda próbuje maksymalizować/minimalizować funkcję celu $f(x)$, gdzie x jest stanem dyskretnym.
- Algorytm będzie przechodził od stanu bieżącego do potomnego lokalnie zwiększając/zmniejszając wartość funkcji f .
- Pierwotnie stosowany z funkcją jednokryterialną.
- Algorytm może być stosowany dla przestrzeni ciągłych i ta jego odmiana jest nazywana metodą gradientową.
- W tym algorytmie przestrzeń problemu jest równa przestrzeni przeszukiwania $\mathbb{G} = \mathbb{X}$.

Metoda największego wzrostu — sposób poszukiwania rozwiązania

Algorytm wykonuje w pętli kroki:

- 1 p najlepsze dotychczas znalezione rozwiązanie produkuje potomka p_{new}
- 2 jeżeli p_{new} jest lepsze od p , to podmienia p

Pierwszy p powstaje w wyniku losowania lub zadanej wartości.

Przykład dla n hetmanów

Rozstawić na szachownicy wielkości $n \times n$, n hetmanów tak, by się nie atakowały.

Założenia

- W każdej kolumnie znajduje się jeden hetman. Zadanie sprowadza się do znalezienia dla każdego z nich odpowiedniego wiersza.
- Procedura generująca potomka p_{new} przesuwa k -tego hetmana do innego wiersza w tej samej kolumnie.
- Funkcja celu $f(p)$ liczy pary atakujących się hetmanów. Gdy $f(p) = 0$, to znaleziono optimum globalne.
- Oczekuje się zmniejszania wartości $f(p)$ w miarę poszukiwania optimum.

Przykład dla 8 hetmanów

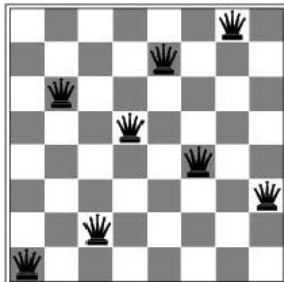
Bieżące rozwiązanie (Rys. lewy)

Wartość funkcji celu $f(p) = 17$. W polach podane są wartości f dla każdego możliwego potomka, który powstanie przez przesunięcie hetmanów w odpowiednie pole.

Stan końcowy (Rys. prawy)

minimum lokalne o wartości $f(p) = 1$.

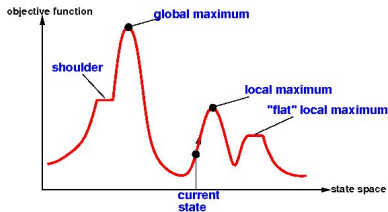
18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♚	13	16	13	16
♚	14	17	15	♚	14	16	16
17	♚	16	18	15	♚	15	♚
18	14	♚	15	15	14	♚	16
14	14	13	17	12	14	12	18



Wady algorytmu największego wzrostu

- Przedwczesna zbieżność (utykanie w lokalnych optimach).
- Obszary płaskie (shoulder i flat local optimum) — obszary przestrzeni przeszukiwania, gdzie na dużym obszarze wartość funkcji celu jest stała lub słabo zmienna. Algorytm ma problem z określeniem kierunku poszukiwania (błądzenie).

Dla 8-hetmanów hill climbing utyka w 86% przypadków, gdy stan początkowy jest losowany.



Warianty algorytmu metody największego wzrostu (1)

Stochastyczna metoda największego wzrostu

Wybiera losowo pomiędzy ścieżkami wznastającymi, przy czym prawdopodobieństwo wyboru zależy od nachylenia zbocza. Im bardziej strome, tym większe prawdopodobieństwo.

Metoda największego wzrostu z listą tabu

Przechowuje potomków niedawno odwiedzonych, by ponownie do nich nie wracać.

Losowe przemieszczenia

Jest to wariant predykcyjny, który na płaskich obszarach próbuje generować potomków w przód (liczba kroków w przód jest z góry zadana) i odszukać narastające zbocza.

Warianty algorytmu metody największego wzrostu (2)

Losowo restartowana metoda największego wzrostu (Shotgun HC)

- Próbuje pokonać problem przedwczesnej zbieżności.
- Iteracyjnie wykonuje HC za każdym razem startując z innego losowego punktu p .
- Najlepszy \hat{p}_l jest przechowywany: jeżeli kolejne uruchomienie wyprodukuje lepsze \hat{p}_l to zostaje ono podmienione.
- Metoda jest zadziwiająco dobra, gdyż zazwyczaj lepiej wykorzystać czas na obliczenia eksplorując szerzej przestrzeń problemu niż próbować optymalizować punkt startu.

Plan wykładu

- 1 Inteligencja obliczeniowa
- 2 Optymalizacja
- 3 Hill climbing
- 4 Symulowane wyżarzanie**
 - Opis algorytmu
 - Wady i zalety
- 5 Literatura

Simulated annealing (SA) (1983)

- Jest probabilistycznym algorytmem przeznaczonym dla zadań globalnej optymalizacji w dużych przestrzeniach stanów.
- Stosowany głównie w przestrzeniach dyskretnych.
- Symuluje proces studzenia kryształów metalu w metalurgii.

Wyżarzanie w metalurgii

- Jest to technika polegająca na kontrolowanym podgrzewaniu i studzeniu materiału, by uzyskać specyficzne jego cechy np. twardość.
- Kryształy metalu mają niewielkie defekty, dyslokacje jonów, które osłabiają całą strukturę. Poprzez ogrzewanie metalu energia jonów (i również współczynnik rozpuszczalności) jest zwiększana. Dyslokacje są niszczone i struktura metalu jest zmieniana w czasie, gdy metal się studzi i osiąga swoją nową równowagę.
- Temperatura początkowa nie może być zbyt niska. Studzenie jest kontrolowane i powinno być wolne, gdyż szybkie prowadzi do osiągnięcia lokalnego optimum funkcji energii (metal).

Fizyka wyżarzania

Każdy zbiór stanów wszystkich atomów w systemie (pos) jest ważony przez współczynnik prawdopodobieństwa Boltzmannna

$$e^{-\frac{E(pos)}{k_B \cdot T}}$$

gdzie:

- $E(pos)$ jest energią konfiguracji pos
- T jest temperaturą podaną w Kelvinach
- $k_B = 1.380650524 \cdot 10^{-23}$ J/K stała Boltzmannna.

Procedura wyżarzania

Czym jest element x ?

Każdy punkt z przestrzeni problemu jest widziany jako stan pewnego fizycznego systemu, a funkcja $E(pos)$ (minimalizowana) jest analogią do stanu energii wewnętrznej systemu. Funkcja E jest funkcją celu.

Cel

Doprowadzić system z arbitralnie wybranego stanu początkowego do stanu z minimalnym stanem energetycznym.

Cykl algorytmu

Podstawowy cykl w każdym kroku rozważa:

- 1 pewnego sąsiada pos_{i+1} konfiguracji bieżącej pos_i , który otrzymuje się przez niewielkie zmiany pos_i .
- 2 Z pewnym prawdopodobieństwem decyduje, czy zmienić stan na pos_{i+1} , czy pozostać w pos_i .
- 3 Prawdopodobieństwo dobierane jest tak, iż ostatecznie system dąży do stanu o mniejszej energii $E(pos)$.
- 4 Zakończenie procesu następuje po uznaniu, że znalezione rozwiązanie jest zadowalające lub gdy wyczerpany zostanie założony limit iteracji.

Procedura generowania sąsiada

Jak tworzyć kolejny stan?

Stany sąsiednie (kandydaci na rozwiązanie) są tworzone zgodnie z pewną z góry określoną procedurą, która działa na konfiguracji bieżącej *pos*;

Np. w problemie n hetmanów jest to przesunięcie hetmanów w kolumnie.

Np. w problemie komiwojażera stan sąsiedni może powstać przez wymianę parami miast.

Prawdopodobieństwo zmiany stanu

$P(\Delta E, T)$

Prawdopodobieństwo przejścia ze stanu pos_i do stanu pos_{i+1} jest określane przez funkcję $P(\Delta E, T)$, która zależy od energii dwóch stanów $E(pos_i)$ i $E(pos_{i+1})$ i globalnego parametru zmiennego w czasie, oznaczonego przez T i nazywanego temperaturą.

$$P(\Delta E, T) = \begin{cases} e^{-\frac{\Delta E}{k_B \cdot T}}, & \text{jeżeli } \Delta E > 0 \\ 1, & \text{w przeciwnym wypadku,} \end{cases}$$

gdzie

$$\Delta E = E(pos_{i+1}) - E(pos_i)$$

Interpretacja wzoru na $P(\Delta E, T)$

- P jest niezerowe, gdy $E(pos_{i+1}) > E(pos_i)$, co oznacza, że system może przejść do stanu gorszego (o wyższym poziomie energetycznym) niż bieżący. Zapewnia to, iż algorytm nie zbiega przedwcześnie do lokalnych optimów.
- Prawdopodobieństwo musi być zmienne w czasie zgodnie z temperaturą studzenia T :
 - Gdy $E(pos_{i+1}) > E(pos_i)$ i $T \rightarrow 0$, to $P(\Delta E, T) \rightarrow 0$.
 - Gdy $E(pos_{i+1}) < E(pos_i)$ i $T \rightarrow 0$, to $P(\Delta E, T)$ rośnie.
 - $T = 0$ algorytm zmienia się w hill climbing.

Funkcja $P(\Delta E, T)$ i jej wpływ na działanie algorytmu.

- Jeżeli $P(\Delta E, T) = 1$, gdy $\Delta E < 0$, to zawsze kieruje się rozwiązanie w stronę mniejszych stanów energetycznych, nie zważając na temperaturę. Nie zawsze musi to być korzystne.
- P jest tak dobrane, by prawdopodobieństwo akceptacji stanu gorszego małało, gdy różnica $E(pos_{i+1}) - E(pos_i)$ rośnie. Zatem małe skoki w kierunku gorszych rozwiązań są bardziej prawdopodobne niż duże.
- Zmiany stanu pos_i są silnie zależne od T . Gdy T jest duże, to zmiany są dużo bardziej agresywne, a gdy T jest małe, to zmiany są niewielkie.

Rozkład wyżarzania

Rozkład temperatury

- Rozkład wyżarzania musi być dany (założony).
- Temperatura w trakcie procesu jest stopniowo zmniejszana. Początkowo T jest dużą wartością i jest zmniejszana z pewnym założonym schematem podanym przez użytkownika. Musi się skończyć z wartością $T = 0$. Np. z każdym krokiem iteracji, jest mnożona przez pewną stałą mniejszą od 1.

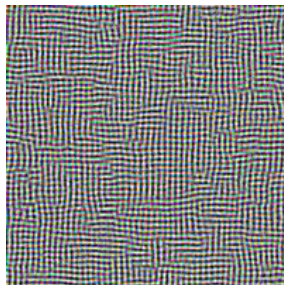
Z tego powodu w początkowej fazie przestrzeń problemu jest badana szeroko (na dużym obszarze) a potem przenosi się w region o niskiej wartości energetycznej i bada okoliczną przestrzeń coraz wężiej, aż zmieni się w algorytm HC.

<http://www.youtube.com/watch?v=KQYfaitQn7g&feature=related>

Przykład



szybkie studzenie



wolne studzenie

Niskie stany energetyczne oznaczają przyciąganie się pikseli o podobnych kolorach, a gdy kolory są różne to zwiększa się ich odległość (stan energetyczny). Analizowano tylko dwa sąsiadujące piksele.

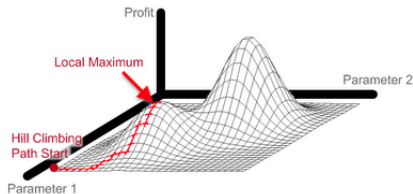
Źródło: http://en.wikipedia.org/wiki/Simulated_annealing

Lista parametrów, na które wrażliwy jest algorytm

- przestrzeń problemu (jak reprezentowani są kandydaci na rozwiązanie)
- funkcja celu: funkcja E
- procedura generowania sąsiadów
- prawdopodobieństwo przejścia ze stanu do stanu P
- schemat rozkładu temperatury
- temperatura początkowa.

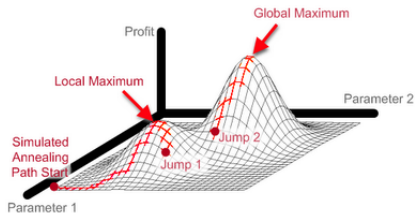
Porównanie działania HC i SA

The problem with hill climbing is that it gets stuck on "local-maxima"

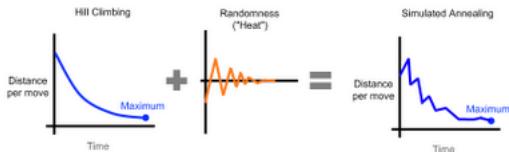


HC

Simulated Annealing can escape local minima with chaotic jumps



SA



Plan wykładu

- 1 Inteligencja obliczeniowa
- 2 Optymalizacja
- 3 Hill climbing
- 4 Symulowane wyżarzanie
- 5 Literatura**

Wykorzystana literatura (do samodzielnego studiowania)



Thomas Weise

Global Optimization Algorithms - Theory and Application.

online e-book: <http://www.it-weise.de/>



S.J. Russel, P. Norvig

Artificial Intelligence. A modern approach.

Pearson Education wyd. 2, p.111-116