Wykład 2

Heurystyki lokalnych ulepszeń

Kazimierz Grygiel

Home Page
Title Page
Page 1 of 25
Go Back
Full Screen
Close
Quit

Jak skonstruować generator rozwiązań?

- Specyficzny generator dla specyficznej klasy funkcji (problemów)
- Rozpoznanie i wykorzystanie strukturalnych regularności, charakteryzujących tę klasę
- Trzy etapy konstrukcji:
  - wprowadzenie (odkrycie) struktury w zbiorze rozwiązań (*sąsiedztwo, podobieństwo, pokrewieństwo, odległość*)
  - powiązanie zależności strukturalnych z ilościowymi (określenie rodzaju regularności)
  - sformułowanie strategii heurystycznego przeszukiwania wykorzystującej hipotetyczne regularności
- *Uwaga:* trzeba pamiętać, że nasze hipotezy będą często *zawodne*; inaczej moglibyśmy je sformułować w postaci twierdzenia i skonstruować na ich podstawie „porządny” algorytm

Sąsiedztwo, przekształcenia modyfikujące

- Podejście strukturalne: rozkład rozwiązania na komponenty
- Odległość rozwiązań: liczba „różnic"
- Sąsiedztwo rozwiązania x:

$$N(x) = \{y \in S : d(x, y) = 1\}$$

- Podejście operacyjne: przekształcenia generujące sąsiedztwo (*przekształcenia modyfikujące, ruchy, mutacje*)

$$\Phi = \{\phi_i : S \to S,\ i \in I\}$$

- Sąsiedztwo rozwiązania x:

$$N(x) = \{\phi_i(x),\ i \in I\}$$

Iterowane przeszukiwanie lokalne — idea

- Ogólna koncepcja:
  - „punkt wypadowy” i penetracja sąsiedztwa
  - podążanie za „tropem”, zmiana punktu wypadowego
- Schemat formalny pojedynczej „rundy” algorytmu
  (heurystyka przeszukiwania z ograniczoną pamięcią)

$$
\begin{array}{c}
x_b, y_b \\
\downarrow \\
\overbrace{\underbrace{\begin{array}{cccc} x_0, & x_1, & \ldots, & x_k \\ y_0, & y_1, & \ldots, & y_k \end{array}}} \\
\downarrow \\
x_b^{'}, y_b^{'}
\end{array}
$$

($x_b, x_b^{'}$ — *rozwiązania bieżące*)

- Sposób wyboru nowego rozwiązania bieżącego określa *reguła selekcji*

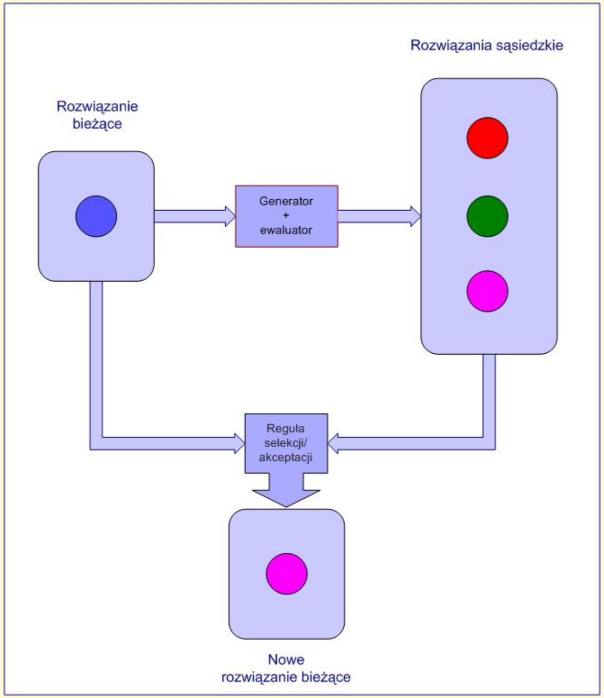
Iterowane przeszukiwanie lokalne — opis

- Wybieramy rozwiązanie początkowe, które staje się *rozwiązaniem bieżącym*
- Generujemy pewną liczbę rozwiązań z sąsiedztwa rozwiązania bieżącego i spośród nich wybieramy kolejne rozwiązanie bieżące
- Powtarzamy ostatni krok pewną liczbę razy, zapamiętując za każdym razem najlepsze otrzymane rozwiązanie
- Jako wynik (aproksymację optimum) przyjmujemy najlepsze zapamiętane rozwiązanie, czyli

$$x_{opt}(K) = \mathsf{opt}_f\{x_0, x_1, \ldots, x_K\}$$

gdzie K: liczba wygenerowanych rozwiązań

Schemat działania



Heurystyki lokalnych ulepszeń

- Jak rozpoznawać „trop" (tzn. jak wybierać nowe rozwiązanie bieżące)?
  - jak zapewnić, żeby rozwiązania bieżące nie powtarzały się, unikając ich rejestracji?
  - jak zapewnić, żeby ciąg rozwiązań bieżących zmierzał do rozwiązania optymalnego?
- *Pomysł*: metoda dojścia na szczyt we mgle — iść stale pod górę
- *Realizacja*: monotoniczna reguła selekcji – podstawa *heurystyki lokalnych ulepszeń*
- Formalnie: heurystyka iterowanego przeszukiwania lokalnego jest heurystyką lokalnych ulepszeń, jeżeli w każdej iteracji rozwiązanie bieżące albo zostaje ulepszone, albo nie ulega zmianie
- Synonim: *algorytm wspinaczki*

Home Page
Title Page
◀◀ ▶▶
◀ ▶
Page 7 of 25
Go Back
Full Screen
Close
Quit

Konsekwencje selekcji monotonicznej

- Co zyskujemy (lub nie)?
  - pierwszy problem rozwiązany całkowicie
  - drugi problem rozwiązany częściowo i niezadowalająco: pułapka lokalnych szczytów
- Formalnym odpowiednikiem „lokalnego szczytu” jest pojęcie *optimum lokalnego*: rozwiązanie x jest *optimum lokalnym*, jeśli w jego sąsiedztwie $N(x)$ nie istnieje rozwiązanie *lepsze*
- *Wniosek*:

  Metoda wspinaczki działa, jeśli jest tylko jeden szczyt; w przeciwnym przypadku może się załamać

- Uwaga: „Teren równinny” (*plateau*) w „krajobrazie” optymalizacyjnym składa się z samych optimów lokalnych!

Home Page
Title Page
Page 8 of 25
Go Back
Full Screen
Close
Quit

Deterministyczne reguły selekcji

- Notacja: $R(x, f, \Phi)$, gdzie

  x: rozwiązanie bieżące

  f: funkcja oceny

  Φ: rodzina przekształceń modyfikujących

- SAHC (Steepest Ascent Hill Climbing):

  $R(x, f, \Phi) :=$ *najlepsze (ze względu na f) rozwiązanie w ciągu*

  $$x, \phi_1(x), \ldots, \phi_{|I|}(x)$$

  (w przypadku niejednoznaczności: pierwsze z nich)

- NAHC (Next Ascent Hill Climbing):

  $R(x, f, \Phi) :=$ *pierwsze (o ile istnieje) rozwiązanie w ciągu*

  $$\phi_1(x), \ldots, \phi_{|I|}(x)$$

  lepsze (ze względu na f) niż x; w przeciwnym przypadku x

Home Page

Title Page

Page 9 of 25

Go Back

Full Screen

Close

Quit

Zrandomizowane heurystyki lokalnych ulepszeń

- Zrandomizowany generator rozwiązań: wybór losowego rozwiązania z sąsiedztwa rozwiązania bieżącego
- Prosta realizacja: *operator modyfikacji* ϕ_ξ, gdzie
  ξ — zmienna losowa o wartościach ze zbioru indeksowego I
  (Inaczej mówiąc, za każdym razem losujemy jedno z przekształceń modyfikujących z rodziny Φ)[1]
- Reguła selekcji redukuje się do *reguły akceptacji*: przyjąć lub odrzucić nowe rozwiązanie
- Konieczne jest *zewnętrzne* kryterium zatrzymania (niemożność sprawdzenia, czy wyczerpano wszystkie elementy sąsiedztwa)

[1] *Uwaga terminologiczna*: *Operatorem* będziemy tu nazywać losowe przekształcenie z danej rodziny przekształceń

Formalizacja

- Relacje „porządku jakościowego”:
  - porządek rosnący

$$x' \prec_f x'' \equiv \begin{cases} f(x') < f(x'') & \text{(maksymalizacja)} \\ f(x') > f(x'') & \text{(minimalizacja)} \end{cases}$$

    (czytamy: x' *jest gorsze niż* x'' *ze względu na* f)
  - porządek malejący

$$x' \succ_f x'' \equiv x'' \prec_f x'$$

    (czytamy: x' *jest lepsze niż* x'' *ze względu na* f)
- Podzbiór lepszych sąsiadów:

$$N^+(x) = \{z \in N(x) : x \prec_f z\}$$

- Rozwiązanie x jest *optimum lokalnym*, jeśli

$$x \not\prec_f z \text{ dla dowolnego } z \in N(x)$$

  Warunek równoważny: $N^+(x) = \emptyset$

Schemat zrandomizowanej heurystyki lokalnych ulepszeń

- Oznaczenia:

  S – zbiór rozwiązań zadania

  ϕ_ξ – operator modyfikacji na S

  $\prec_f$ – relacja porządku jakościowego na S (ze względu na funkcję f)

- Algorytm:

```
begin
    x := losowy element z S;
    repeat
        z := φ_ξ(x);
        if x ≺_f z  then x := z
    until warunek zatrzymania;
    return x
end
```

Czas oczekiwania na poprawę rozwiązania

- Algorytm: zrandomizowana heurystyka lokalnych ulepszeń
- Oznaczenia:

  $p_s(x) = \mathbf{P}\left\{\phi_\xi(x) \in N^+(x)\right\}$

  $q_s(x) = 1 - p_s(x)$

  $T_s(x)$: czas oczekiwania na poprawę rozwiązania

- Wówczas

$$\mathbf{P}\left\{T_s(x) = t\right\} = q_s^{t-1}(x) p_s(x)$$

(rozkład geometryczny)

$$\mathbf{E} T_s(x) = \begin{cases} 1/p_s(x) & \text{jeśli } p_s(x) > 0 \\ +\infty & \text{w p. p.} \end{cases}$$

Charakteryzacja wyników

- Jakie rozwiązanie znajduje heurystyka lokalnych ulepszeń (ogólniej: monotoniczna)?
- Wariant deterministyczny:
  - przestrzeń rozwiązań rozpada się na rozłączne *obszary przyciągania* optimów lokalnych
  - wynik zależy wyłącznie od punktu startowego
  - optima lokalne = punkty stałe reguły selekcji R
- Wariant zrandomizowany:
  - nie ma obszarów przyciągania
  - algorytm zachowuje się jak *pochłaniający łańcuch Markowa*
  - optima lokalne = stany pochłaniające
- Kiedy heurystyki monotoniczne znajdują optimum globalne niezależnie od punktu startowego?
- Gdy wszystkie optima lokalne są globalne (inaczej: gdy lokalne ulepszenia zawsze zbliżają do celu)

Uogólnione struktury sąsiedztwa

- Skąd biorą się optima lokalne?
- Odpowiedź: sami je wprowadziliśmy, ograniczając przeszukiwanie do sąsiedztwa rozwiązania bieżącego
- Jak złagodzić ten problem?
- Hierarchia sąsiedztw:

  $$k\text{-sąsiedztwo rozwiązania } x = \{y \in S : \ \rho(x,y) \leqslant k\},$$

  gdzie $\rho(\cdot,\cdot)$ — odległość (metryka) w S
- Mechanizm uprzywilejowania *bliskiego* sąsiedztwa: stopniowe oddalanie się od punktu wyjścia
- Heurystyki zrandomizowane: *stochastyczna lokalność* operatora; prawdopodobieństwo malejące wraz ze wzrostem odległości („zasięgu" operatora)

Reprezentacje: przestrzeń wektorów zerojedynkowych

- Przestrzeń wektorów zerojedynkowych

$$B_m = \{x = (x_1, x_2, \ldots, x_m),\ \text{gdzie } x_j \in \{0, 1\}\}$$

z metryką Hamminga: $d_H(x, y) = \sum_{j=1}^{m} \mid x_j - y_j \mid$

- Przekształcenia modyfikujące: *mutacje punktowe*

$$\Phi = \{\phi_j(x_1, \ldots, x_j, \ldots, x_m) = (x_1, \ldots, 1{-}x_j, \ldots, x_m),\ j \in I\}$$

gdzie $I = \{1, 2, \ldots, m\}$

- *Operator mutacji punktowej* ϕ_ξ:

gdzie ξ — zmienna losowa o rozkładzie jednostajnym na I, tj.

$$\mathbf{P}\{\xi = j\} = \frac{1}{m} \text{ dla } j \in I$$

Home Page
Title Page
Page 16 of 25
Go Back
Full Screen
Close
Quit

Przykład: analiza czasu oczekiwania na optimum

- $p_s(x) = \frac{|N^+(x)|}{|N(x)|}$ dla $x \in B_m$, $|N(x)| = m$
- funkcja oceny: $f(x) = \sum_{i=1}^{m} x_i$, zadanie maksymalizacji
- optimum globalne: $(1, 1, \ldots, 1)$ (brak innych optimów lokalnych)
- rozwiązanie startowe: $x^0 = (0, 0, \ldots, 0)$
- $p_s(x^k) = (m-k)/m$, więc

$$\mathbf{E}T_s(x^k) = m/(m-k) \text{ dla } k=0, 1, \ldots m-1$$

- całkowity średni czas oczekiwania:

$$m \sum_{k=0}^{m-1} \frac{1}{m-k} = mH_m \cong m \ln(m)$$

Home Page
Title Page
Page 17 of 25
Go Back
Full Screen
Close
Quit

Mutacja Bernoulliego

- Mutacje wielopunktowe (wg maski):

$$\Phi = \{\phi_i(x) = x \oplus i,\ \ i \in B_m\}$$

gdzie i — *maska mutacyjna*, $\oplus$ — dodawanie modulo 2 po współrzędnych

- Operator mutacji Bernoulliego ϕ_ξ:
ξ — wektor losowy o wartościach z B_m, przy czym

$$\mathbf{P}\{\xi = i\} = p^{/i/}(1-p)^{m-/i/}$$

gdzie p — *punktowe prawdopodobieństwo mutacji*
$/i/$ — liczba jedynek w masce i

- Średni zasięg operatora (oczekiwana liczba zmutowanych pozycji w wektorze x): mp
- Rozsądne wartości: $0<p<0.5$
- Uwaga: $p = 0.5$ generuje rozkład jednostajny na S

Mutacja Bernoulliego: objaśnienia

- Jak działa maska mutacyjna?

x	0	0	1	1
i	0	1	0	1
$x \oplus i$	0	1	1	0
p_i	q	p	q	p

- *Interpretacja*: Jedynka w masce oznacza zajście mutacji punktowej na danej pozycji
- *Prawdopodobieństwo maski* (dla tego przykładu): p^2q^2 $(q = 1 - p)$
- Uwaga końcowa: Mutacja Bernoulliego z $p < 0.5$ jest przykładem operatora stochastycznie lokalnego!

Reprezentacje binarne: implementacja

- Implementacja tablicowa: jeden bajt na zapamiętanie jednego bitu (spore marnotrawstwo, ale bez ograniczeń)
- Implementacja całkowitoliczbowa: możliwa w zakresie umiarkowanych długości reprezentacji
- Przykładowa implementacja w Pascalu:

```
const
    L = 30; { długość chromosomu }
    N = Integer(1) shl L; { N = 2^L }
type
    Chromosom = 0..(N-1);
```

- Operacje na bitach
  - dodawanie wektorowe modulo 2 (operacja $\oplus$): `xor`
  - przesunięcie o jeden bit w lewo / w prawo: `shl` / `shr`

„Zegar mutacyjny”

- Dla małych wartości p można zastosować efektywną implementację operatora mutacji Bernoulliego, opartą na następującym twierdzeniu:

  Jeżeli p jest prawdopodobieństwem sukcesu dla nieograniczonego ciągu prób Bernoulliego oraz U — zmienną losową o rozkładzie jednostajnym na odcinku $[0,1]$, to $K = \lfloor \ln(U)/\ln(1-p) \rfloor$ jest zmienną losową reprezentującą liczbę porażek przed pierwszym sukcesem

- Zamiast dokonywać próby losowej dla każdej pozycji wektora, losujemy miejsca, w których zachodzi mutacja punktowa
- Gdy mutacje są rzadkie, daje to znaczną oszczędność czasu

Reprezentacje: przestrzeń wektorów rzeczywistoliczbowych

- Przestrzeń wektorów rzeczywistoliczbowych

$$\mathcal{R}^m = \{x = (x_1, x_2, \ldots, x_m),\ \text{gdzie } x_j \in \mathcal{R}\}$$

  z metryką euklidesową lub miejską

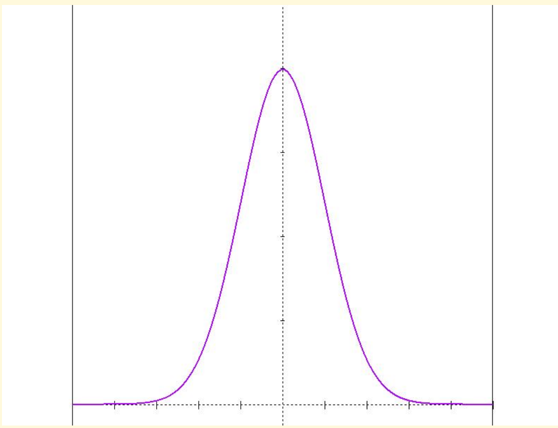
- Przekształcenia modyfikujące: mutacje addytywne: $\phi_u(x) = x + u$, gdzie $x, u \in \mathcal{R}^m$
- Operator mutacji gaussowskiej $\phi_\xi(x) = x + \xi$:
  ξ — wektor losowy o m-wymiarowym rozkładzie normalnym o średniej $\mathbf{0}$ i macierzy kowariancji $\mathbf{C}$.
- Wariant uproszczony $\mathbf{C} = \mathbf{D}^2$ (brak korelacji)
- Wtedy $F(x_1, \ldots, x_m) = \prod_{i=1}^{m} F_i(x_i)$,

  gdzie $F_i(x) = (\sigma_i\sqrt{2\pi})^{-1} \int_{-\infty}^{x} e^{-\frac{t^2}{2\sigma_i^2}} dt$

Home Page
Title Page
Page 22 of 25
Go Back
Full Screen
Close
Quit

Rozkład normalny (jednowymiarowy)

- Notacja: $\xi \sim N(a, \sigma)$ — zmienna losowa ξ ma rozkład normalny z parametrami a, σ
- Parametry: a: wartość oczekiwana, σ: odchylenie standardowe (σ^2 – wariancja)
- Funkcja gęstości: $g(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(t-a)^2}{2\sigma^2}}$ (*krzywa Gaussa*)



Charakterystyka mutacji gaussowskiej

- Wariant standardowy: mutacje niezależne dla każdej współrzędnej rozwiązania
- Rozkład zasięgu mutacji ($\mathbf{P}\{|\xi| < k\sigma\}$):

odchylenie	% przypadków
$0,67\sigma$	50,0
σ	68,3
2σ	95,6
3σ	99,7

- Średni zasięg (oczekiwana długość „kroku mutacyjnego"):

$$\mathbf{E}|\xi| = \sigma\sqrt{\frac{2}{\pi}} \cong 0,80\sigma$$

- Uwaga: mutacja gaussowska jest także przykładem operatora stochastycznie lokalnego

Genererowanie liczb losowych o rozkładzie normalnym

- Rozkład $N(0,1)$ (metoda Boxa-Mullera):

  Jeżeli U_1, U_2 są niezależnymi zmiennymi losowymi o rozkładzie jednostajnym na przedziale (0,1), to

  $$X_1 = \sqrt{-2\ln U_1}\cos(2\pi U_2)$$

  $$X_2 = \sqrt{-2\ln U_1}\sin(2\pi U_2)$$

  są niezależnymi zmiennymi losowymi o rozkładzie normalnym $N(0,1)$

- Rozkład $N(a,\sigma)$:

  $$Y = a + \sigma X$$

  gdzie X – zmienna losowa o rozkładzie $N(0,1)$

Home Page
Title Page
Page 25 of 25
Go Back
Full Screen
Close
Quit